(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 221 651 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
10.07.2002 Bulletin 2002/28

(51) Int Cl.[7]: **G06F 9/445**

(21) Application number: **01130280.9**

(22) Date of filing: **20.12.2001**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **09.01.2001 JP 2001001858**

(71) Applicant: **SEIKO EPSON CORPORATION**
Shinjuku-ku, Tokyo 163-0811 (JP)

(72) Inventors:
- **Kobayashi, Atsushi**
  Suwa-shi, Nagano-ken 392-8502 (JP)
- **Shizuka, Sachie**
  Suwa-shi, Nagano-ken 392-8502 (JP)
- **Sugimoto, Toshiyuki**
  Suwa-shi, Nagano-ken 392-8502 (JP)

(74) Representative: **Hoffmann, Eckart, Dipl.-Ing.**
Patentanwalt,
Bahnhofstrasse 103
82166 Gräfelfing (DE)

(54) Device environment configuration system, device environment configuration method, and data storage medium therefor

(57) A DEC (Device Environment Configuration) system and method define a device profile containing setup information for a device according to how the device is connected to the system by displaying device profiles according to how the various devices in the system are connected. An input device (402) inputs device profiles for setting the OPOS-DEC settings to match how the device are connected in the POS terminal system. A device profile editor (403) relates the input data to a device profile previously stored in the OPOS-DEC settings so that a particular device profile can be edited. The device profile verification unit (404) verifies whether the edited profile is appropriate based on the OPOS device environment settings. Based on the edited profile data, the device profile updating unit (405) then updates the OPOS-DEC settings in the registry (410). The device profile display controller (406) displays device properties based on the OPOS device environment settings.

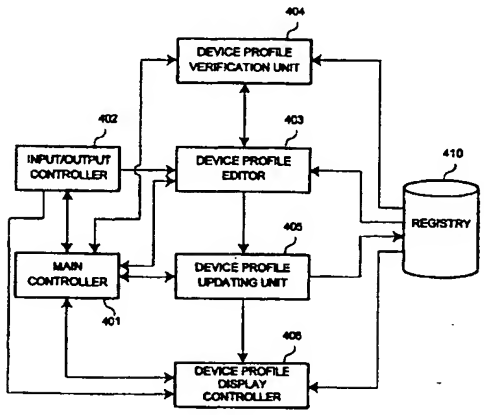


FIG. 4

EP 1 221 651 A2

Description

[0001] The present invention relates generally to a DEC (Device Environment Configuration) system, a DEC method, and a data storage medium carrying computer program means for implementing the method.

[0002] A multifunction POS (point-of-sale) system can be assembled without the user being aware of the makes and models of devices in the POS system configuration by using a standard Application Program Interface (API) to develop an application providing the POS system functions. OPOS (OLE for Retail POS) is an international standard defining a standard interface for devices (such as printers and displays) used in the POS terminals of a POS system (referred to below as a POS terminal system). An OPOS standard object uses OLE (Object Linking and Embedding) control supported by the Windows (R) OS used on the host devices of the POS system to provide application developers with a set of API functions that make application development easier.

[0003] The information needed to use OPOS, that is, information about all of the devices in the POS terminal system and settings for all of the devices (referred to below as the OPOS-DEC settings), is stored in a database (referred to below as a registry) of system configuration information managed by the Windows OS. This registry is a database for centrally managing all settings related to computer operation, including all device driver settings and application settings. Device information including a device identifier and various settings stored in the registry as part of the OPOS device environment configuration settings (OPOS-DEC settings) is referred to below as the device information or the device profile. In other words, the OPOS-DEC settings include device information for each device in the system configuration.

[0004] It is usually necessary to use a registry editor, utility software for editing the registry, in order to save OPOS-DEC settings for a device in the POS terminal system to the registry. The Windows OS provides a registry editor utility, called regedit.exe, for this purpose.

[0005] The registry editor, however, only provides simple basic functions. This makes editing the OPOS-DEC settings for each device in the POS terminal system and saving the settings to the registry a cumbersome, time-consuming task. More specifically, it is difficult to edit and save the OPOS-DEC settings while also considering the configuration of all other devices in the POS terminal system. Yet further, it is difficult to determine and confirm the appropriateness of the edited OPOS-DEC settings so that the appropriate settings can be saved in the registry. It is also difficult to edit and save the OPOS-DEC settings with consideration for how the different devices in the POS terminal system are connected to each other.

[0006] As noted above the registry is a central database of system configuration information managed by directly by the OS, making it difficult for someone without sufficient knowledge to edit the registry. Furthermore, inappropriately changing registry content could also render the OS, applications, or devices unusable or interfere with normal operation. Inadvertently changing data other than the OPOS-DEC settings could also cause problems with the system.

[0007] This invention is therefore directed to a solution for these problems, and an object of this invention is to provide a DEC system having means for displaying device profiles included in the system configuration data managed by the OS based on how the devices in the system are connected and means for writing device settings appropriate to how a device is connected in a device profile of the system configuration data.

[0008] This object is achieved with a system as claimed in claim 1 and a method as claimed in claim 16. Preferred embodiments of the invention are subject-matter of the dependent claims.

[0009] Searching for solutions for the prior art problems described above, the inventors realized that the POS system developer can easily understand how devices in the POS terminal system are connected by displaying a device tree organized by, for example, device class and connection port based on the OPOS-DEC settings stored in the registry. They showed that a detailed profile for a selected device can be displayed through a simple operation by using an object properties function.

[0010] The inventors also showed that OPOS-DEC settings can be easily edited by, for example, using a set-up wizard to add all required settings to the device profiles of the devices in the POS terminal system; batch adding information for all devices in a POS terminal system integrating a host computer with peripheral devices; changing device information using a drag-and-drop function to simply move all device information for the device connected to the port where dragging starts so that the device is reconnected to the port where the device information is dropped; and changing or deleting information for a specified device and the information for all devices connected to the specified device based on how the specified device is connected.

[0011] A DEC system according to the present invention achieving the above objects has a device information setting means for displaying, based on the connection status of one or more devices in the system, device profiles from the system configuration information managed by the system OS in desired display groups and means for writing a device profile for a particular device to the system configuration information (i.e., the registry) so that the profile is appropriate to the device connection status.

[0012] The method according to the present invention can be provided as a computer-executable program, and the program can be recorded on and distributed as data storage medium such as a Compact Disc (CD), a floppy disc, a hard disc, a magneto-optical disc, a Digital Versatile Disc, a magnetic tape, or a semiconductor memory.

[0013] Other objects and attainments together with a fuller understanding of the invention will become apparent and appreciated by referring to the following description of preferred embodiments taken in conjunction with the accompanying drawings, in which:

Fig. 1 shows the configuration of a POS terminal system;

Fig. 2 is a block diagram of the POS terminal system;

Fig. 3 (a) shows where OPOS is positioned in a POS terminal system, and (b) is a flow chart of system development for a POS terminal system;

Fig. 4 is a function block diagram for a DEC system;

Fig. 5 is a flow chart of the DEC process;

Fig. 6 is a flow chart of the display routine in the DEC process;

Fig. 7 is a flow chart of the add routine in the DEC process;

Fig. 8 is a flow chart of the delete routine in the DEC process;

Fig. 9 is a flow chart of the change information routine in the DEC process;

Fig. 10 is a flow chart of the routine for handling other operations in the DEC process;

Fig. 11 is an example of a device class display screen;

Fig. 12 (a) is an example of connection port display tree, and (b) shows a pass-through connection and a Y connection;

Fig. 13 (a) shows selecting the verification process, and (b) shows a warning displayed by the verification process;

Fig. 14 (a) shows a standard view of device information, and (b) shows a name view in alphabetic order;

Fig. 15 is an example of a screen showing detailed device information using the device property function;

Fig. 16 shows a device class selection screen;

Fig. 17 shows a device selection screen;

Fig. 18 shows a screen for setting communication parameters;

Fig. 19 shows a screen for adding a logical device name;

Fig. 20 shows a screen for batch registering devices;

Fig. 21 shows deleting a device that is not hydra connected;

Fig. 22 shows deleting a hydra connected device; and

Fig. 23 (a) shows device connections before changing device connection port information with a drag-and-drop function, and (b) shows device connections after the change.

[0014] Fig. 1 shows the configuration of a typical POS terminal system 100. This POS terminal system 100 comprises a host computer 101, printer 102, customer display 103, and cash drawer 104. The host computer 101 stores a software application providing various functions of the POS terminal system 100. The host computer 101 is connected by a communication bus 110 to the printer 102 for printing receipts and a transaction journal, for example, the customer display 103 for displaying purchase information, and the cash drawer 104 for holding money and checks. Various communication protocols can be used on the communication bus 110, including serial, parallel, and network (such as IEEE (Institute of Electrical and Electronic Engineers) 802).

[0015] Fig. 2 is a block diagram of a host computer 101. The host computer 101 has an interface 201 for connecting to the communication bus 110, a disk drive 202 for reading software and information from a CD (Compact Disc) or other medium, a CPU 203, RAM 204 for storing the program run by the CPU 203 and used as working memory by the CPU 203, a hard disk 205 storing the program and data used by the CPU 203, a monitor 206 or other type of display device for presenting messages and information for the user, a floppy disk drive 207 for reading software and other information from a floppy disk, a keyboard and mouse or other type of input device 208 enabling user input to the system, and a bus 210 connecting these various components.

[0016] Fig. 3 illustrates the role of OPOS in a POS terminal system. Fig. 3 (a) shows where OPOS is positioned, and Fig. 3 (b) is a flow chart of system development for a POS terminal system. As shown in Fig. 3 (a), OPOS is located between the OS and the application controlling the devices in the POS terminal system, and provides a standardized interface for connecting the

host computer (PC) and peripheral devices based on defined specifications.

[0017] OPOS has two layers, the control object (CO) and service object (SO) layers. The control object layer contains objects provided for each device class, such as printers and displays, and controls the device-application interface. The service object layer contains service objects provided for each device, such as for each printer model. The service objects control device operation by way of the OS. The OS also manages the registry where the OPOS-DEC settings used by OPOS are stored.

[0018] The application controls a particular device using methods and properties, and receives feedback about the result from the controlled device using events and properties. OPOS converts process requests from the application to commands supported by the device and sends the commands to the addressed device, and receives the device status indicating the process result from the device.

[0019] As shown in Fig. 3 (b) building a POS terminal system involves four major tasks. First is installing OPOS to the host computer used for development (step S301). This computer is referred to as the "developer station". More specifically, this involves installing the control object and service object, and creating an OPOS-DEC settings layer in the registry; the device class is then saved in this layer. The OPOS-DEC settings are then edited to match the devices used in the POS terminal system, and saved in the registry of the developer station (step S302). More specifically, a device profile is registered below the previously registered device class. A software application providing the desired functions of the POS terminal system is then developed on the developer station (step S303). The last step is to reproduce the application environment and application developed on the developer station on the host computers of the plural POS terminal systems of the POS system (these host computers are referred to below as client PCs) (step S304), thus completing the POS system.

[0020] A DEC system enabling the task shown as step S302 in the construction of a POS terminal system to be completed more efficiently, that is, the task of editing the OPOS-DEC settings to match the devices used in the POS terminal system and saving these settings in the registry, is described below according to the present invention.

[0021] Fig. 4 is a function block diagram of this DEC system. The various functions are described below. As shown in Fig. 4 this DEC system has a main controller 401, input device 402, device profile editor 403, device profile verification unit 404, device profile updating unit 405, and device profile display controller 406.

[0022] The input device 402 inputs the device profiles for setting the OPOS-DEC settings in the registry 410 to match the connections of the devices in the POS terminal system. The device profiles are input using a keyboard, mouse, or other input device not shown in the figures.

[0023] The device profile editor 403 relates the data input by the input device 402 to a device profile, if any, previously stored in the OPOS-DEC settings in the registry 410, and thus edits individual device profiles in the OPOS DEC.

[0024] The device profile verification unit 404 determines whether the device profile edited by the device profile editor 403 is appropriate based on the OPOS-DEC settings in the registry 410. If the profile is not appropriate, a warning is passed to the display controller 406 and displayed, and the device profile editor 403 is instructed to re-edit the edited device settings until the data is appropriate.

[0025] Based on the device profile edited by the device profile editor 403, the device profile updating unit 405 then updates the OPOS-DEC settings in the registry 410.

[0026] On a display device not shown in the figures, the display controller 406 presents a device tree organized by device class or port based on the OPOS-DEC settings in the registry 410, and, using the property function, displays device profiles, including detailed information about a device and various settings. The display controller 406 can also display a map of device connections (hydra connection). As further described below, a hydra connection is a method for connecting plural devices to a single port.

[0027] The main controller 401 controls and arbitrates operation of the input device 402, device profile editor 403, device profile verification unit 404, device profile updating unit 405, and display controller 406.

[0028] A summary of the DEC system process is described next below with reference to the flow chart in Fig. 5. When the program for the DEC process starts, the first step is to display the main screen for configuring the device environment (step S501), and then detect what input was received (step S502).

[0029] If the input is to display a device profile (step S502 returns "DISPLAY"), the OPOS-DEC settings in the registry 410 are displayed according to the input data (step S503), and the procedure loops back to step S501 to display the main screen and wait for the next input. If the input was to display the device profiles organized by device class, for example, the devices are displayed grouped by the device class, such as printers and displays, as shown in Fig. 11.

[0030] If the input is to add device information (step S502 returns "ADD"), the device information input via the input device is related to the previously registered OPOS-DEC settings in the registry 410 so that the settings for the particular device can be edited (step S504), the settings are then verified for applicability (step S505), a process for adding the edited data to the OPOS-DEC settings in the registry 410 is run (step S506), and the procedure loops back to step S501 to display the main screen and wait for the next input.

[0031] If the input is to delete device information (step S502 returns "DELETE"), the device information input from the input device is related to the previously registered OPOS-DEC settings in the registry 410 and edited (step S507), then verified (step S508), and a process for deleting the information from the OPOS-DEC settings in the registry 410 is then run (step S509). Next, the procedure loops back to step S501 to display the main screen and wait for the next input.

[0032] If the input is to change device information (step S502 returns "CHANGE"), the device information input from the input device is related to the previously registered OPOS-DEC settings in the registry 410 and edited (step S510), then verified (step S511), and a process for changing the OPOS-DEC settings in the registry 410 based on the changed data is run (step S512). Next, the procedure loops back to step S501 to display the main screen and wait for the next input.

[0033] If the input is for an operation other than to edit device information as described above (step S502 returns "OTHER"), the corresponding process is run (step S513), and the procedure then loops back to step S501 to display the main screen and wait for the next input.

[0034] Data is verified in steps S505, S508, and S511 by selecting a verification process as shown in Fig. 13 (a). When the verification process detects a problem, a screen such as shown in Fig. 13 (b) is displayed. The user can choose to use an auto-correct function to automatically change the setting to a valid setting. The user can also deselect the verification process so that this process is not run.

[0035] The steps in the DEC process are described in further detail below. Fig. 6 is a flow chart of the display process shown as step S503 in Fig. 5.

[0036] The first step (S601) is to determine whether to display device connections, a list of all devices, or a device profile (device details and settings). If displaying device connections is selected (S601 returns "DEVICE TREE"), what type of tree to display is determined (S602).

[0037] If a device class tree display is selected (S602 returns "DEVICE CLASS"), that is, if the user selects Device Class View from the view menu of the main screen, the connected devices are displayed grouped by device class (S603). More specifically, the tree groups printers together, groups display devices together, and so forth. A typical device class tree is shown in Fig. 11. Device classes for which device information cannot be set (such as devices that are not provided but belong to an OPOS standard device class) can be indicated by changing the color or shade used to display those device classes.

[0038] The order of the device class display can be switched between a standard view arranging the device classes in a predetermined sequence (such as from highest to lowest importance to POS system configuration), and an alphabetic sequence by device class name. The desired display sequence can be specified using a menu selection (standard order, order by name), for example.

[0039] Fig. 14 (a) shows devices displayed in a standard order, and (b) shows the devices arranged by name. Note that Fig. 11 and Fig. 14 show the trees in expanded form, but the tree is collapsed when the device class tree or port tree is first displayed.

[0040] If the user selects a connection port tree view (S602 returns "PORT"), that is, if the user selects Port Connection View from the view menu of the main screen, the ports are searched to detect the hydra connection map from the device information (S604).

[0041] As mentioned before, a hydra connection is a method of connecting plural devices to a single port, and can take two forms: pass-through connection or Y connection. Fig. 12 (b) shows the physical arrangement of pass-through and Y connections. With a pass-through connection data flows from the host computer (PC) to the receive buffer of the customer display (LineDisplay), and only the data for the printer (POS printer) is then passed from the LineDisplay to the printer. With a Y connection the data is sent simultaneously from the host computer to both the customer display and printer.

[0042] The devices are then displayed based on the device connection data detected for each port, that is, in order of physical proximity to the host computer (S605). Fig. 12 (a) shows a typical display of the devices organized by connection port. The hydra connection of devices to serial interface ports COM1 and COM2 can be easily visually confirmed.

[0043] If displaying all devices is selected (S601 returns "ALL DEVICES"), that is, the user selects View All from the view menu, all device classes and all devices in the POS terminal system in the device class or connection port display trees shown by steps S602 to S605, are displayed (S606). More specifically, this option expands all branches of the device tree as shown in Fig. 11 and Fig. 14.

[0044] If displaying device properties including details about and settings for the devices is selected (S601 returns "PROPERTIES"), that is, the user selects Property from the right-click menu of the selected device, a dialog box showing the settings of the selected device is displayed in a tab format using the object property function (S607). An example of the device property dialog box is shown in Fig. 15. This example shows the properties for the customer display device DM-D101.

[0045] Fig. 7 is a flow chart of the routine for adding information (S504 in Fig. 5) as part of the DEC process. This add routine is run when selected from the edit menu of the main screen or from the right-click menu of a device class or device. The first step is determining what device information to add (S701). When Add New Device is selected from the edit menu of the main screen (S701 returns "ADD DEVICE PROFILE"), device information is added using a wizard. In this case a device class selection screen is presented (S702), a device selection screen is presented (S703), and a device com-

munication parameter configuration screen is presented (S704) to prompt the user for information. A device profile is then added based on the information input to the displayed screens. It will be noted that a wizard is a method of prompting the user for information while also offering guidance concerning what operations to perform.

[0046] Fig. 16 shows a typical device class selection dialog box, Fig. 17 shows a device selection dialog box, and Fig. 18 shows a device communication parameter dialog box. Using these dialog boxes the user selects, for example, the "printer" device class, selects the printer model, and then inputs the printer-specific communication parameters to add device information.

[0047] The wizard can also provide a device function test called "Check Health" (such as test print if the device is a printer) and set up device-specific parameters (such as set the paper size if the device is a printer) as shown in Fig. 18.

[0048] A device can also be added without using a wizard. If a device class tree is displayed in the main screen and the user selects adding a device by right-clicking on the device class of the device to add, the device class selection screen is skipped so that the user can start from the device selection dialog box. For example, if "printer" is selected as the device class from the main screen and the user then right clicks to add a device, adding device information can start from the device selection screen for printers.

[0049] If the connection port tree is displayed in the main screen, the user selects the port to which the device is connected, and then right clicks to add a device, the device class selection screen must be displayed but the default settings for the selected port can be shown in the device selection screen. For example, if a USB port is selected in the main screen as the port to which a device will be connected, and "add device" is selected by right clicking on the port, "USB" is indicated as the default port in the device selection screen.

[0050] If the user selects adding a Logical Device Name for a device (S701 returns "ADD LOGICAL DEVICE NAME") as a result of the user selecting Add New LDN from the edit menu of the main screen, or selects the device for which to add a name and then right clicks and selects Add New LDN, a dialog box for adding a logical device name is displayed. This dialog box prompts the user for device name input, and then adds the name to the selected device (S705). Fig. 19 shows a typical dialog box for adding a device name.

[0051] If the user selects batch registering devices to a POS terminal integrating a host computer with printer, and other peripheral devices (S701 returns "BATCH REGISTER") by selecting Batch Settings from the edit menu of the main screen, a screen for batch registering all devices connected to the POS PC is displayed. This screen prompts the user to input the configuration, and then adds the device information based on the user-defined settings (S706). A typical dialog box for batch registering plural devices is shown in Fig. 20.

[0052] Fig. 8 is a flow chart of the delete routine (S507 in Fig. 5) in the DEC process. The delete routine is run by selecting Delete from the edit menu of the main screen after first selecting the device to delete, or right clicking on the device and selecting Delete from the pop-up menu. The first step in this routine is to determine if the selected device is a hydra connected device based on the connection profile of the selected device (S801). If the device is a hydra connected device (S801 returns *Yes*, whether the device is a parent or child is detected (S802). If it is a parent device (S802 returns *Yes*, the user is asked whether to delete all connected devices including the child devices (S803).

[0053] If the user says to delete all devices (S803 returns *Yes*, all child devices connected to the selected parent device are found (S804), the device information for all identified devices is deleted (S805), and the device information for the selected parent device is then deleted (S806).

[0054] If deleting all devices connected to the selected device is not selected (S803 returns *No*, only the device information for the selected device is deleted (S807). In this case the children connected to the deleted parent device are connected directly to the port to which the parent device was connected, and this new connection status is reflected in the device information, which is updated for each child device.

[0055] If the hydra connected device is a child device (S802 returns *No*, the device information for the selected device is deleted (S807). If the selected device is not hydra connected (S801 returns *No*, the device information for the selected device is deleted (S807).

[0056] Fig. 21 shows a typical screen for deleting a device that is not hydra connected, and Fig. 22 shows a screen for deleting a hydra connected device. As shown in Fig. 21, plural devices can be selected and all selected devices can then be deleted. As shown in Fig. 22, a parent device can be selected to delete both the parent and all children connected thereto.

[0057] Fig. 9 is a flow chart of the routine for changing (editing) device information shown as step S510 of the DEC process in Fig. 5. The first step is to evaluate the input for changing the registered device information (S901). If device port information is changed by drag-and-drop editing (S901 returns "DRAG-AND-DROP"), the device information for all devices connected to the port where dragging started is found (S902). The device information for all devices found connected to that port is then changed according to the port where the object is dropped (S903).

[0058] Fig. 23 shows an example of changing port information by dragging a device from one port and dropping it on another. Fig. 23 (a) shows the connections before the port is changed, and Fig. 23 (b) shows the devices connected to each port after the device is dropped on a different port to change the connection.

[0059] As will be known from Fig. 23, changing the

port to which LineDisplay (a parent device) is connected from COM1 to COM3 also changes the connection for the child devices connected to the LineDisplay, that is, POSPrinter and CashDrawer, from COM1 to COM3. It should be noted that cut and paste editing can be used instead of drag-and-drop. If cut and paste is used the cut and paste operations can be controlled from the edit menu or using keyboard shortcuts.

[0060] If the selected device is a child, it is possible to move just the child, and if a device that can be a child is moved to a port to which a device that can be a parent is connected, then the child device can be hydra connected to the parent.

[0061] If the communication settings of the selected device are to be changed (S901 returns "SET COMMUNICATION PARAMETERS") and the user selects Communication Settings from the edit menu of the main screen or by right clicking on a device object, a communication settings dialog box is displayed (substantially the same as the dialog box shown in Fig. 18). The user can then change the communication settings for the selected device (S904). It is also possible as part of this process to verify the communication settings and enter device-specific settings.

[0062] If the logical device name of the selected device is to be changed (S901 returns "CHANGE LOGICAL DEVICE NAME") by selecting Rename LDN from the edit menu of the main screen or right clicking on the selected device, a dialog box for changing the logical device name is displayed (substantially the same as the screen shown in Fig. 17). The user can then input the desired name (S905) to change the logical device name of the selected device.

[0063] Fig. 10 is a flow chart for handling processes other than those described above within the DEC process. The first step is to evaluate the input (S1001). If confirming the communication settings is indicated (S1001 returns "CONFIRM") as a result of selecting Check Health Interactive from the tools menu of the main screen, for example, a routine to check for illegal port connections, illegal communication settings, and whether communication is possible using the communication settings is run (S1002). It is also possible to prevent illegal port connections when a device is added to a specific port or a device is changed to a different connection port by confirming the connection settings, such as whether a hydra connection is used. Devices can also be connected without any such confirmation. Connection to illegal ports can also be prevented by defining the ports that can be used.

[0064] If the input is a version compatibility setting (S1001 returns "VERSION COMPATIBILITY") as a result of selecting Version Compatibility from the options menu of the main screen, for example, settings are made to absorb differences in operation resulting from different versions of the application controlling various OPOS functions (S1003). For example, if the application was written for an older version of OPOS but can still be used under a newer version of OPOS with changed or new functions, settings can be made to operate according to the old version.

[0065] If an installer or uninstaller is started (S1001 returns "INSTALL/UNINSTALL"), and Change Component to change the configuration is selected from the options menu of the main screen, an installer/uninstaller is run to change the configuration or change the version of an OPOS function by adding, changing, or deleting OPOS functionality (S1004).

[0066] If a command is applied to save the registry (S1001 returns "SAVE REGISTRY") by selecting Export Registry from the tools menu of the main screen, for example, the OPOS-DEC settings are exported from the registry and saved to a data storage medium (S1005). This task is needed because the OPOS-DEC settings are needed to install the OPOS functions developed on the developer station onto a client PC.

[0067] If the input command is to detect the USB number (S1001 returns "FIND USB DEVICE NUMBER"), the USB number automatically assigned when a device is connected to a USB port is detected to relate the device and the USB number (S1006). Note that this function can also be applied to other communication protocols.

[0068] If the debugger is selected (S1001 returns "DEBUGGER") by selecting OPOS API Trace from the options menu of the main screen, the trace function can be toggled on and off for saving OPOS method and property values to a specific file (S1007).

[0069] If the program functions of this DEC system are assigned to steps in the flow chart of the DEC process shown in Fig. 5, the device information display step corresponds primarily to steps S501 and S503; the input step to step S502; the device profile editing step to steps S504, S507, and S510; the device profile verification step to steps S505, S508, and S511; and the device profile update step to steps S506, S509, and S512. The add device information step corresponds primarily to step S504; the delete device information step to step S507; and the change device information step to step S510.

[0070] A general purpose, practical DEC method can be achieved by incorporating the program functions of the DEC system described above into a computer.

[0071] It is noted that the present invention is not limited to the OPOS devices and can be applied to all other devices for using a computer.

[0072] Some advantages of the present invention are described below.

[0073] How devices in a POS terminal system are interconnected can be visually understood by the POS system developer by displaying a device tree organized by device class, connection port, or other category based on the OPOS-DEC settings stored in the registry. Detailed information about a selected device can also be displayed by a simple operation using the property function.

[0074] OPOS-DEC settings can be easily edited by, for example, using a set-up wizard to add all required settings to the device profiles of the devices in the POS terminal system; batch adding information for all devices in a POS terminal system integrating a host computer with peripheral devices; changing device information using a drag-and-drop function to simply move all device information for a device connected to the port where dragging starts to the port where the device information is dropped; and changing or deleting information for a specified device and the information for all devices connected to the specified device based on how the specified device is connected.

[0075] In other words, while observing the settings and setup of all devices in a POS terminal system, the user can edit the OPOS-DEC settings, verify the appropriateness of the edited OPOS-DEC settings, and save the confirmed, edited content to the registry. OPOS-DEC settings can also be edited according to how the devices of the POS terminal system are connected to each other.

[0076] Mistakenly changing registry data other than the OPOS-DEC settings can also be prevented.

[0077] The time required to develop an application providing functions for a POS terminal system is thus reduced, and an environment enabling efficient application development can be provided for system developers.

Claims

1. A system for configuring a device environment, comprising:

   device information display means (406) for displaying, in desired display groups and based on the connection status of one or more devices in the system, device profiles included in system configuration information (410) managed by the system OS; and
   device information setting means for writing a device profile for a device to the system configuration information so that the profile is appropriate to the device connection status.

2. The system of claim 1, wherein the device information setting means comprises:

   input means (402) for inputting information to be written to a device profile in the system configuration information (410) based on how the devices displayed by the device information display means (406) are connected;
   device profile editing means (403) for editing the device profile based on information input by the input means (402);
   device profile verification means (404) for verifying the device profile edited by the device profile editing means (403); and
   device profile updating means (405) for updating a device profile in the system configuration information (410).

3. The system of claim 2, wherein the device information display means (406) comprises means for displaying one or more devices in the system in a device tree format according to desired display groups.

4. The system of claim 3, wherein the display groups enable devices to be displayed by device class.

5. The system of claim 3, wherein the display groups separate the devices by connection port, and display devices by connection status.

6. The system of claim 2, wherein the device information display means (406) comprises a means for displaying device information for a desired device categorized by a specific parameter.

7. The system of claim 2, wherein the device profile editing means (403) comprises:

   (c1) a device profile generator for adding a new device profile including a device identifier specifying a particular device in the system, and device communication parameters;
   (c2) device profile changing means for changing the device identifier and device communication parameters set in the device profile added to the system configuration information (410) by the device profile generator; and
   (c3) device profile deleting means for deleting the device profile added by the device profile generator from the system configuration.

8. The system of claim 7, wherein the device profile generator is adapted to add the new device profile for a specific device in a specific sequence of operating steps.

9. The system of claim 7 or 8, wherein the device identifier specifying a particular device added by the device profile generator includes a logical device name for the device.

10. The system of any one of claims 7 to 9, wherein the device profile generator is adapted to add in a batch plural devices to a system integrating plural devices of different device classes.

11. The system of any one of claims 7 to 10, wherein the device profile deleting means is adapted to delete all devices connected to a specific device

based on the connection status of the specific device.

12. The system of any one of claims 7 to 11, wherein the device profile changing means is adapted to change in a batch the connection of a specific device connected to a port from which it is moved and the connection of all devices connected thereto in a specific way to the port to which the specific device is moved.

13. The system of any one of claims 7 to 12, wherein the device profile changing means can change only a particular specific setting for a device.

14. The system of claim 13, wherein the specified setting is a device communication parameter.

15. The system of claim 13, wherein the specified setting is the logical device name of the device.

16. A method of configuring a device environment comprising:

(a) writing a device profile for a device to system configuration information (410) so that the profile is appropriate to the device connection status by displaying, based on the connection status of one or more devices in the system, device profiles from the system configuration information (410) managed by the system OS in desired display groups.

17. The method of claim 16, wherein step (a) comprises:

(b) displaying the device profiles from the system configuration information (410) in desired display groups;
(c) receiving input of information to be written to a device profile in the system configuration information (410) based on how the devices displayed by step (b) are connected;
(d) editing the device profile based on information input in step (c);
(e) verifying the device profile edited in step (d); and
(f) updating a device profile in the system configuration information (410).

18. The method of claim 17, wherein step (b) comprises displaying one or more devices in the system in a device tree format according to desired display groups.

19. The method of claim 18, wherein the display groups enable devices to be displayed by device class.

20. The method of claim 18, wherein the display groups separate the devices by connection port, and display devices by connection status.

21. The method of claim 17, wherein step (b) comprises displaying device information for a desired device categorized by a specific parameter.

22. The method of claim 17, wherein step (d) comprises:

(d1) adding a new device profile including a device identifier specifying a particular device in the system, and device communication parameters;
(d2) changing the device identifier and device communication parameters set in the device profile added to the system configuration information (410) by step (d1); and
(d3) deleting the device profile added by step (d1) from the system configuration.

23. The method of claim 22, wherein step (d1) adds settings for a specific device in a specific sequence operating steps.

24. The method of claim 22 or 23, wherein the device identifier specifying a particular device added by step (d1) includes a logical device name for the device.

25. The method of any one of claims 22 to 24, wherein step (d1) can batch add devices to a system integrating plural devices of different device classes.

26. The method of any one of claims 22 to 25, wherein step (d3) can delete all devices connected to a specific device based on the connection status of the specific device.

27. The method of any one of claims 22 to 26, wherein step (d2) can batch change the connection of a specific device connected to a port from which it is moved and the connection of all devices connected thereto in a specific way to the port to which the specific device is moved.

28. The method of any one of claims 22 to 27, wherein step (d2) can change only a particular specific setting for a device.

29. The method of claim 28, wherein the specified setting is a device communication parameter.

30. The method of claim 28, wherein the specified setting is the logical device name of the device.

31. A computer-readable data storage medium carry-

ing computer program means capable of implementing the method as described in any one of claims 16 to 30.

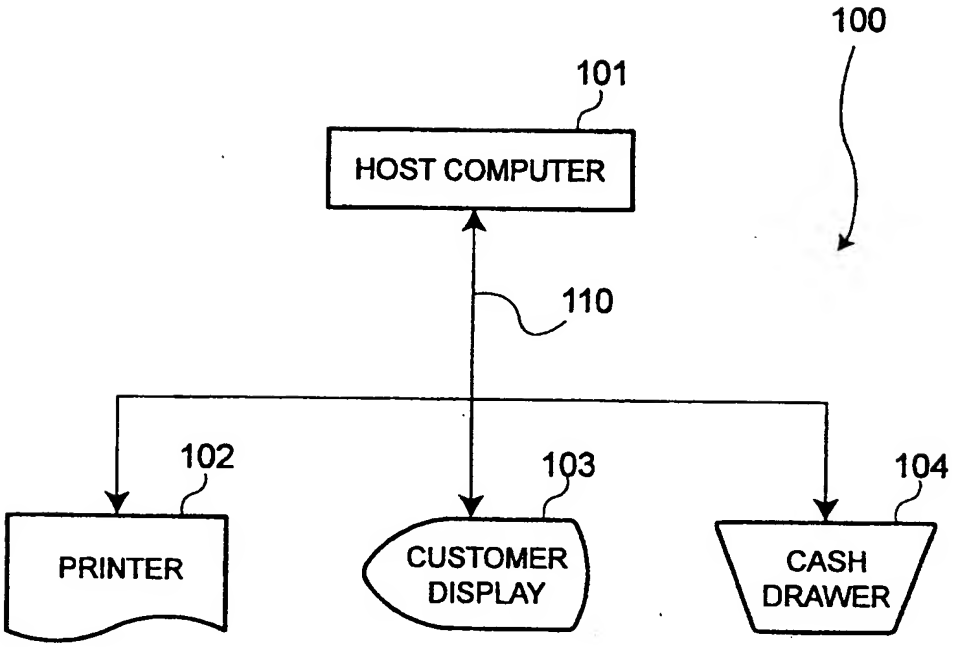

100
101
HOST COMPUTER
110
102
PRINTER
103
CUSTOMER DISPLAY
104
CASH DRAWER

FIG. 1

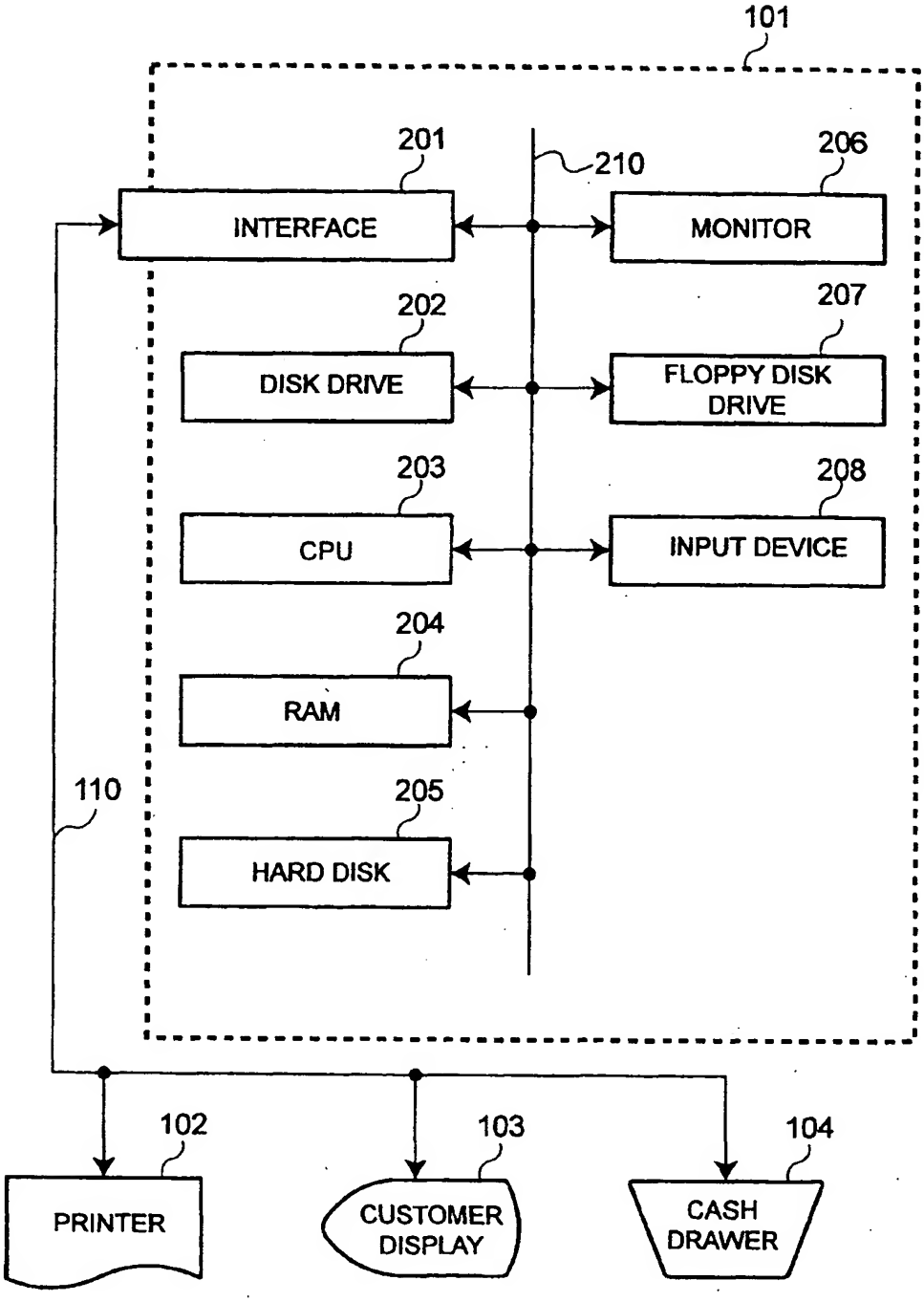

101
201
210
206
INTERFACE
MONITOR
202
207
DISK DRIVE
FLOPPY DISK DRIVE
203
208
CPU
INPUT DEVICE
204
RAM
110
205
HARD DISK
102
103
104
PRINTER
CUSTOMER DISPLAY
CASH DRAWER

FIG. 2

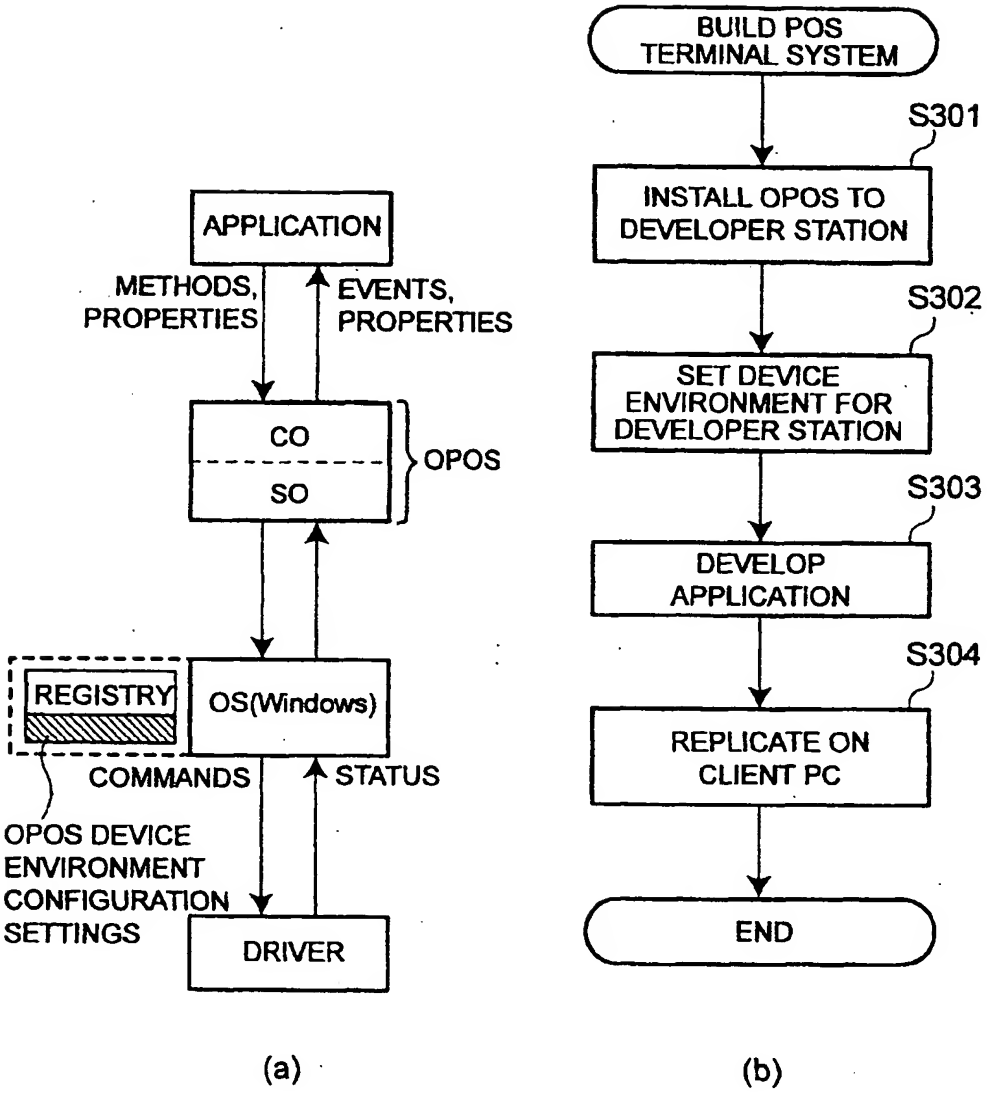

APPLICATION
METHODS, PROPERTIES
EVENTS, PROPERTIES
CO
SO
OPOS
REGISTRY
OS(Windows)
COMMANDS
STATUS
OPOS DEVICE ENVIRONMENT CONFIGURATION SETTINGS
DRIVER
(a)
BUILD POS TERMINAL SYSTEM
S301
INSTALL OPOS TO DEVELOPER STATION
S302
SET DEVICE ENVIRONMENT FOR DEVELOPER STATION
S303
DEVELOP APPLICATION
S304
REPLICATE ON CLIENT PC
END
(b)

FIG. 3

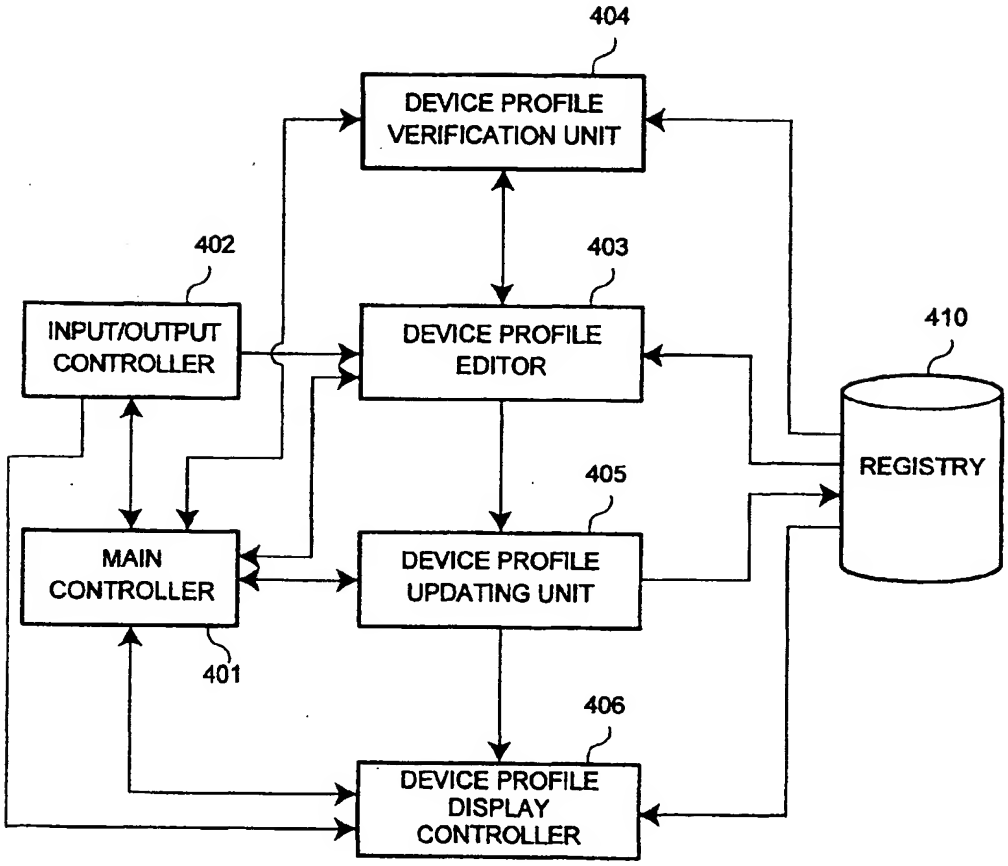

404
DEVICE PROFILE VERIFICATION UNIT
402
403
INPUT/OUTPUT CONTROLLER
DEVICE PROFILE EDITOR
410
REGISTRY
405
MAIN CONTROLLER
DEVICE PROFILE UPDATING UNIT
401
406
DEVICE PROFILE DISPLAY CONTROLLER

FIG. 4

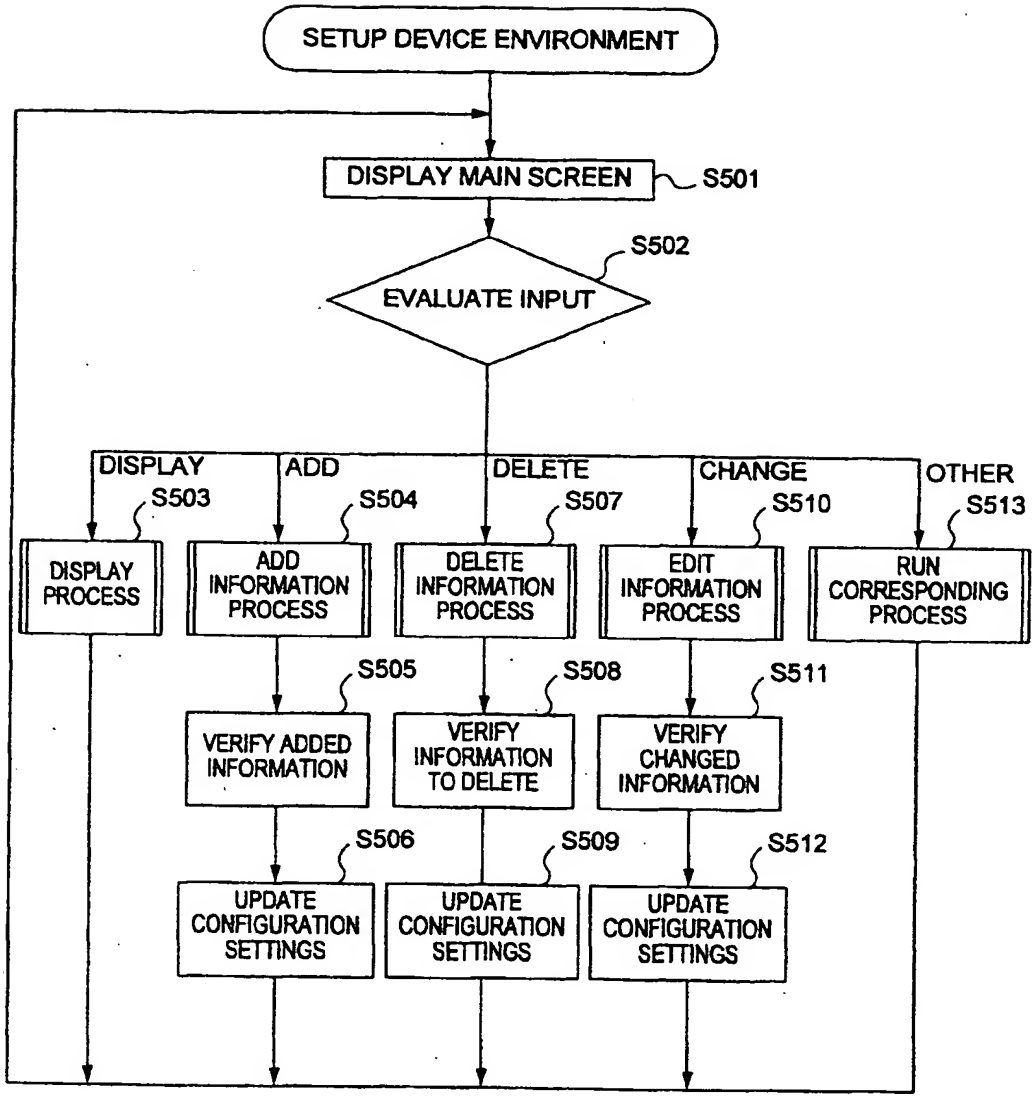

SETUP DEVICE ENVIRONMENT
DISPLAY MAIN SCREEN
S501
S502
EVALUATE INPUT
DISPLAY
ADD
DELETE
CHANGE
OTHER
S503
S504
S507
S510
S513
DISPLAY PROCESS
ADD INFORMATION PROCESS
DELETE INFORMATION PROCESS
EDIT INFORMATION PROCESS
RUN CORRESPONDING PROCESS
S505
S508
S511
VERIFY ADDED INFORMATION
VERIFY INFORMATION TO DELETE
VERIFY CHANGED INFORMATION
S506
S509
S512
UPDATE CONFIGURATION SETTINGS
UPDATE CONFIGURATION SETTINGS
UPDATE CONFIGURATION SETTINGS

FIG. 5

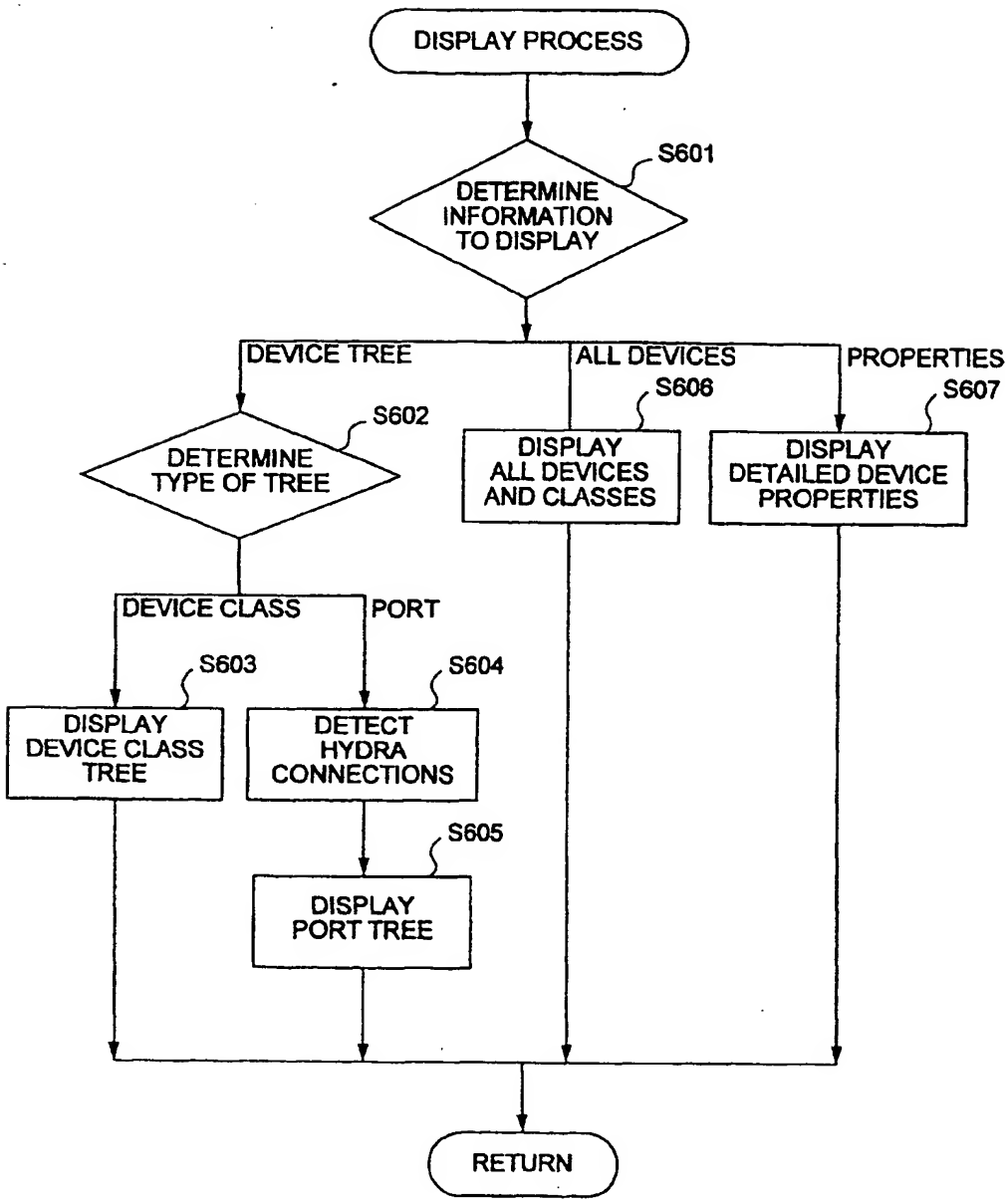

DISPLAY PROCESS
S601
DETERMINE INFORMATION TO DISPLAY
DEVICE TREE
ALL DEVICES
PROPERTIES
S602
DETERMINE TYPE OF TREE
S606
DISPLAY ALL DEVICES AND CLASSES
S607
DISPLAY DETAILED DEVICE PROPERTIES
DEVICE CLASS
PORT
S603
DISPLAY DEVICE CLASS TREE
S604
DETECT HYDRA CONNECTIONS
S605
DISPLAY PORT TREE
RETURN

FIG. 6

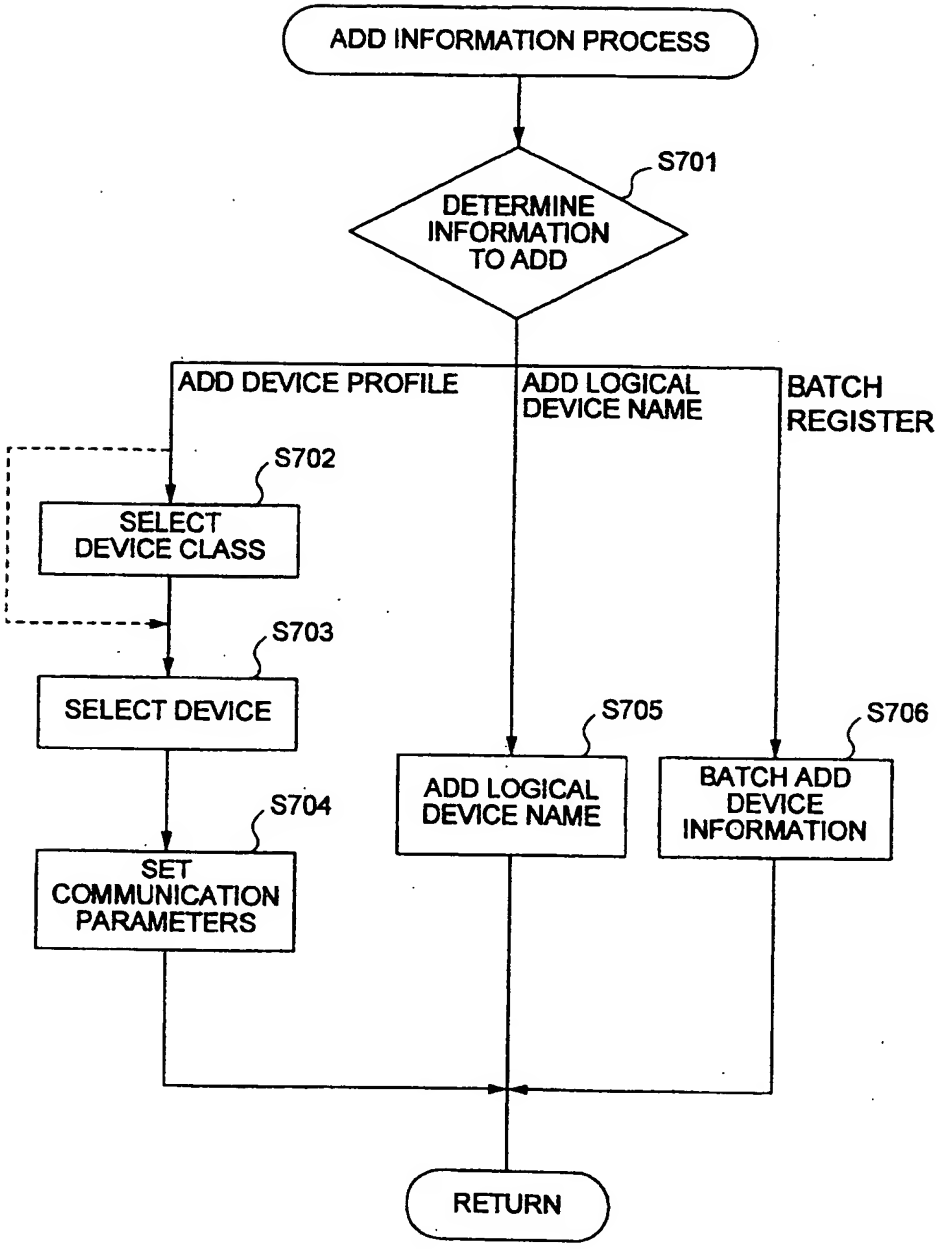

ADD INFORMATION PROCESS
S701
DETERMINE INFORMATION TO ADD
ADD DEVICE PROFILE
ADD LOGICAL DEVICE NAME
BATCH REGISTER
S702
SELECT DEVICE CLASS
S703
SELECT DEVICE
S704
SET COMMUNICATION PARAMETERS
S705
ADD LOGICAL DEVICE NAME
S706
BATCH ADD DEVICE INFORMATION
RETURN

FIG. 7

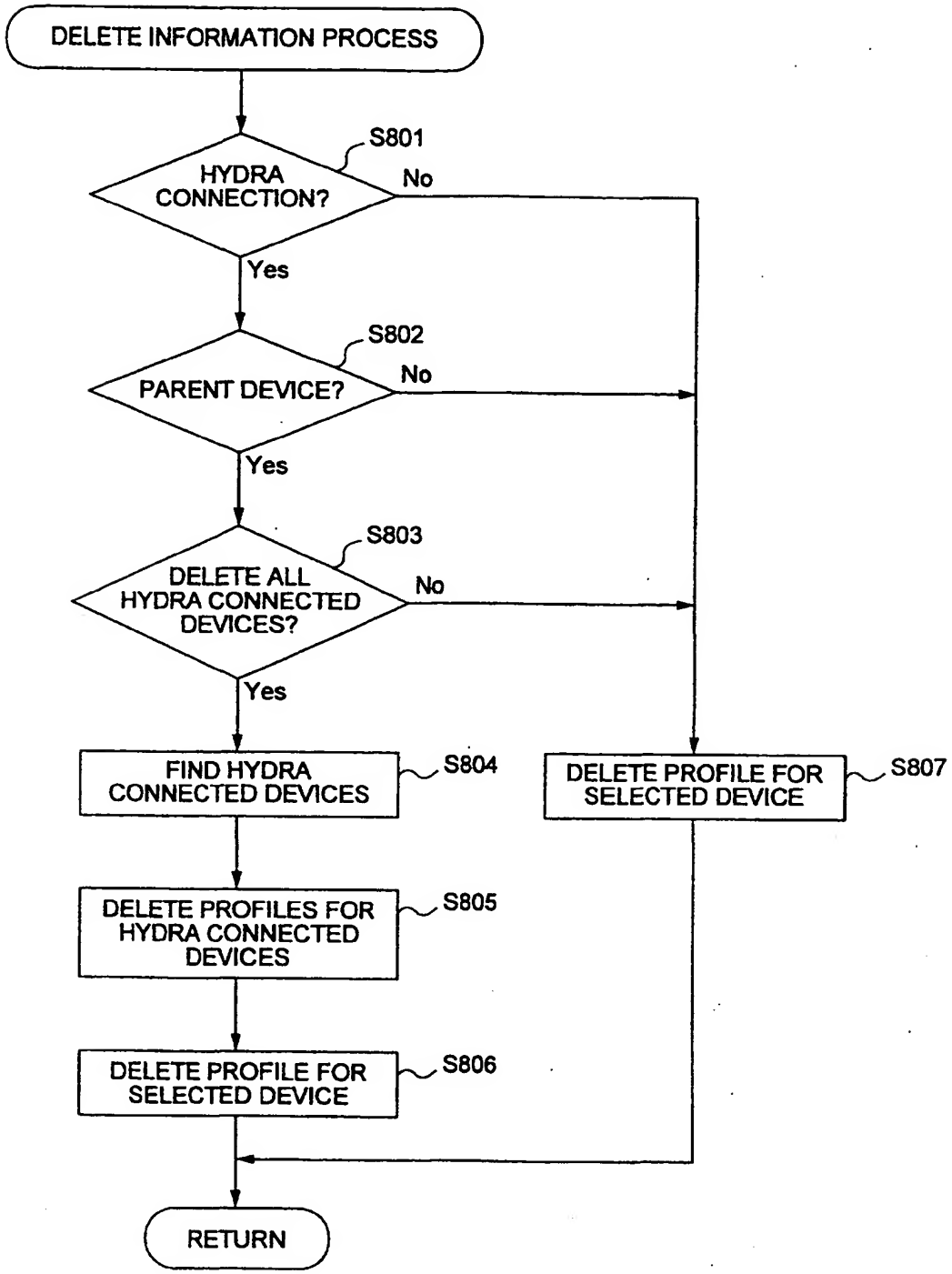

DELETE INFORMATION PROCESS
S801
HYDRA CONNECTION?
No
Yes
S802
PARENT DEVICE?
No
Yes
S803
DELETE ALL HYDRA CONNECTED DEVICES?
No
Yes
FIND HYDRA CONNECTED DEVICES
S804
DELETE PROFILES FOR HYDRA CONNECTED DEVICES
S805
DELETE PROFILE FOR SELECTED DEVICE
S806
DELETE PROFILE FOR SELECTED DEVICE
S807
RETURN

FIG. 8

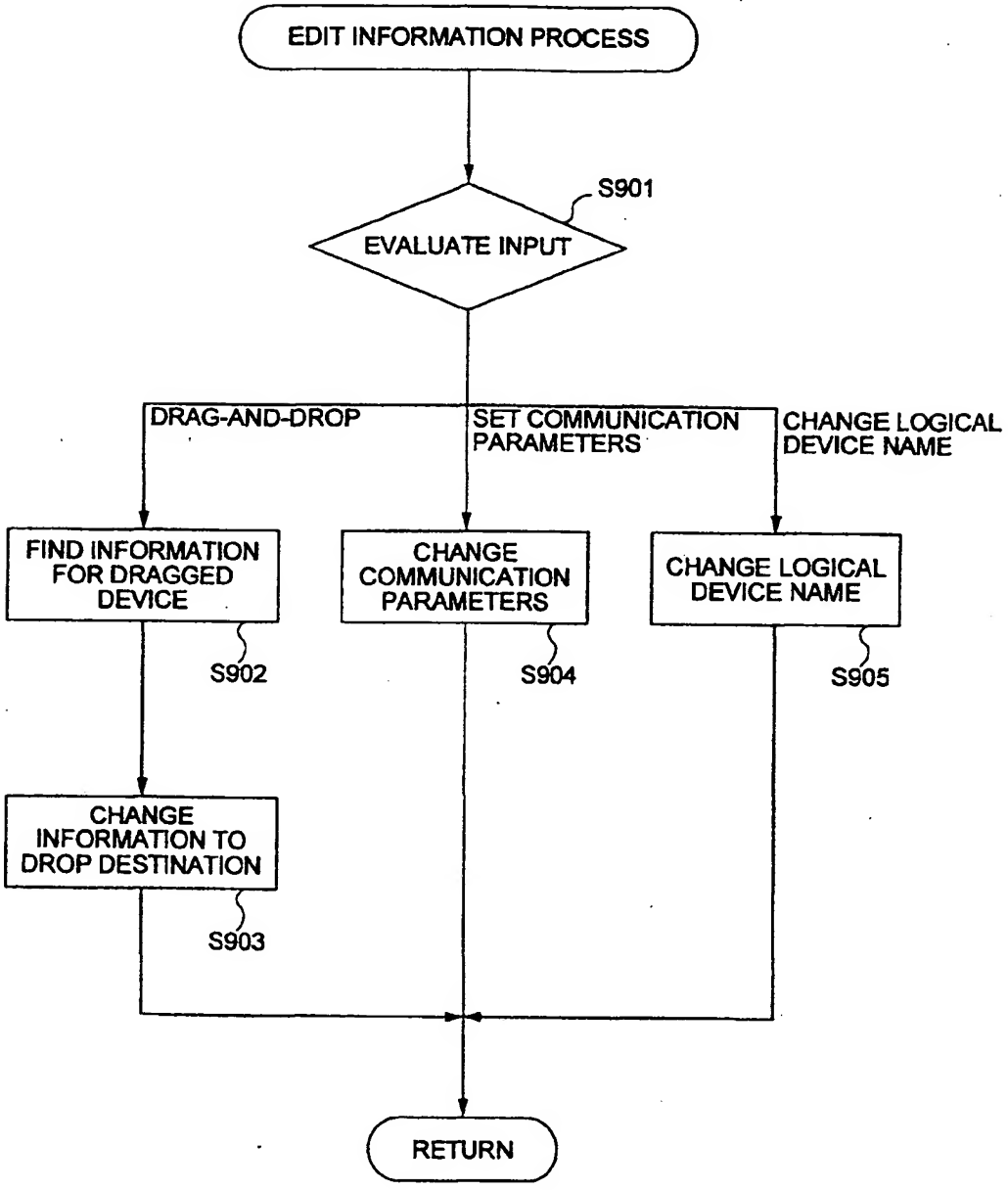

EDIT INFORMATION PROCESS
S901
EVALUATE INPUT
DRAG-AND-DROP
SET COMMUNICATION PARAMETERS
CHANGE LOGICAL DEVICE NAME
FIND INFORMATION FOR DRAGGED DEVICE
CHANGE COMMUNICATION PARAMETERS
CHANGE LOGICAL DEVICE NAME
S902
S904
S905
CHANGE INFORMATION TO DROP DESTINATION
S903
RETURN

FIG. 9

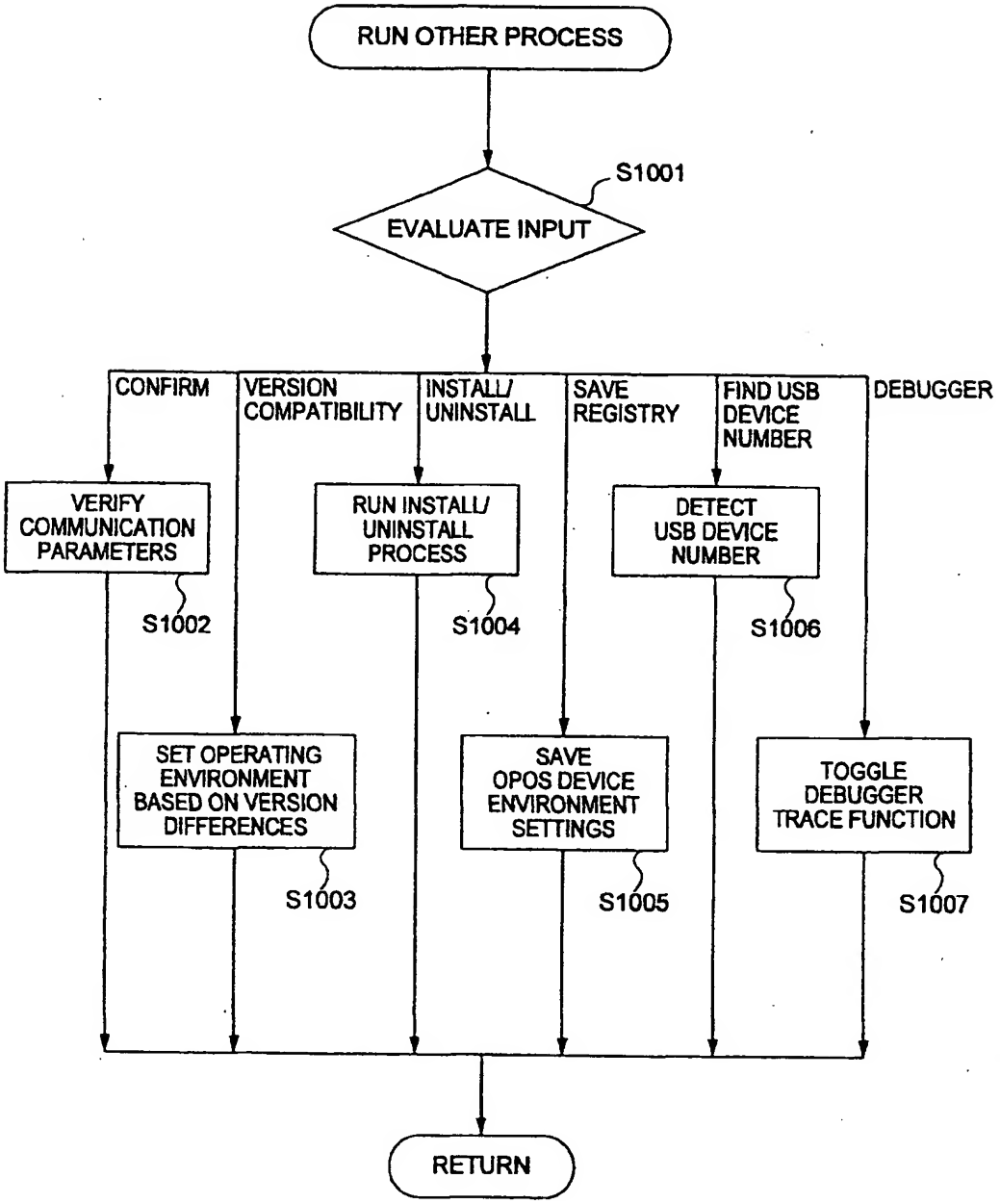

RUN OTHER PROCESS
S1001
EVALUATE INPUT
CONFIRM
VERSION COMPATIBILITY
INSTALL/ UNINSTALL
SAVE REGISTRY
FIND USB DEVICE NUMBER
DEBUGGER
VERIFY COMMUNICATION PARAMETERS
S1002
RUN INSTALL/ UNINSTALL PROCESS
S1004
DETECT USB DEVICE NUMBER
S1006
SET OPERATING ENVIRONMENT BASED ON VERSION DIFFERENCES
S1003
SAVE OPOS DEVICE ENVIRONMENT SETTINGS
S1005
TOGGLE DEBUGGER TRACE FUNCTION
S1007
RETURN

FIG. 10

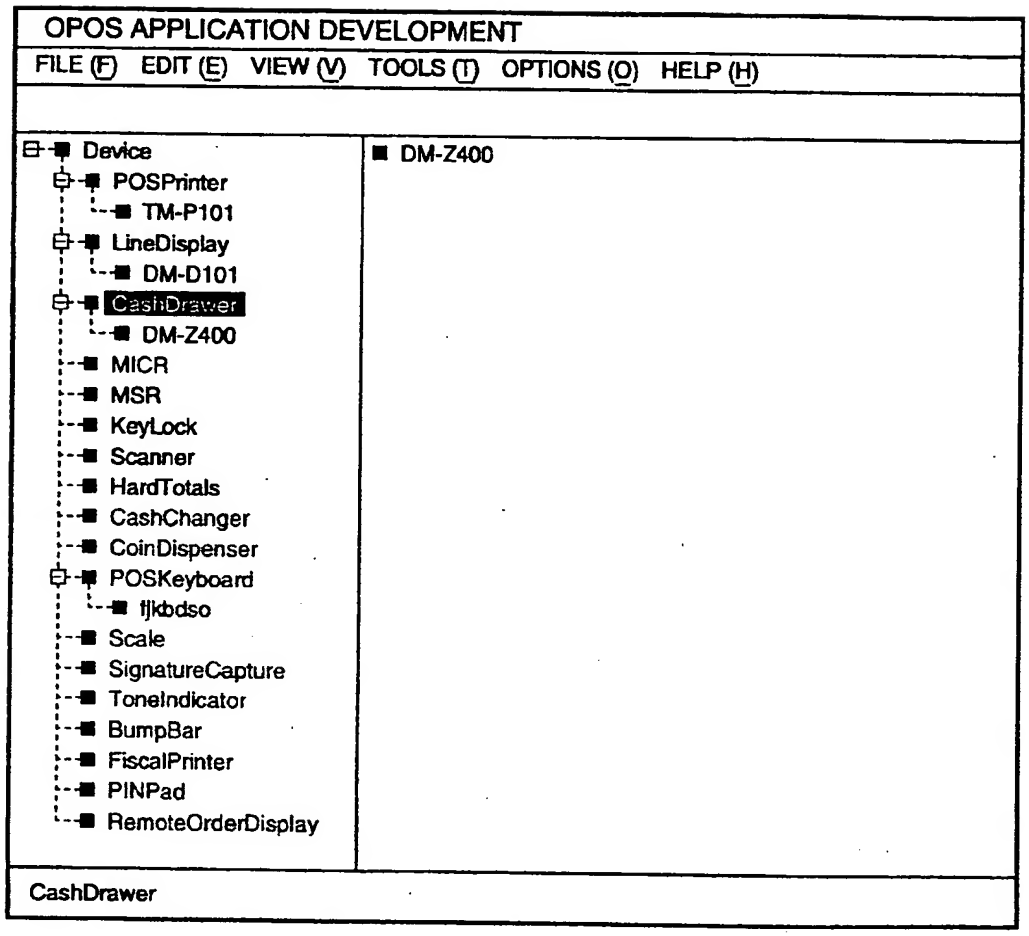
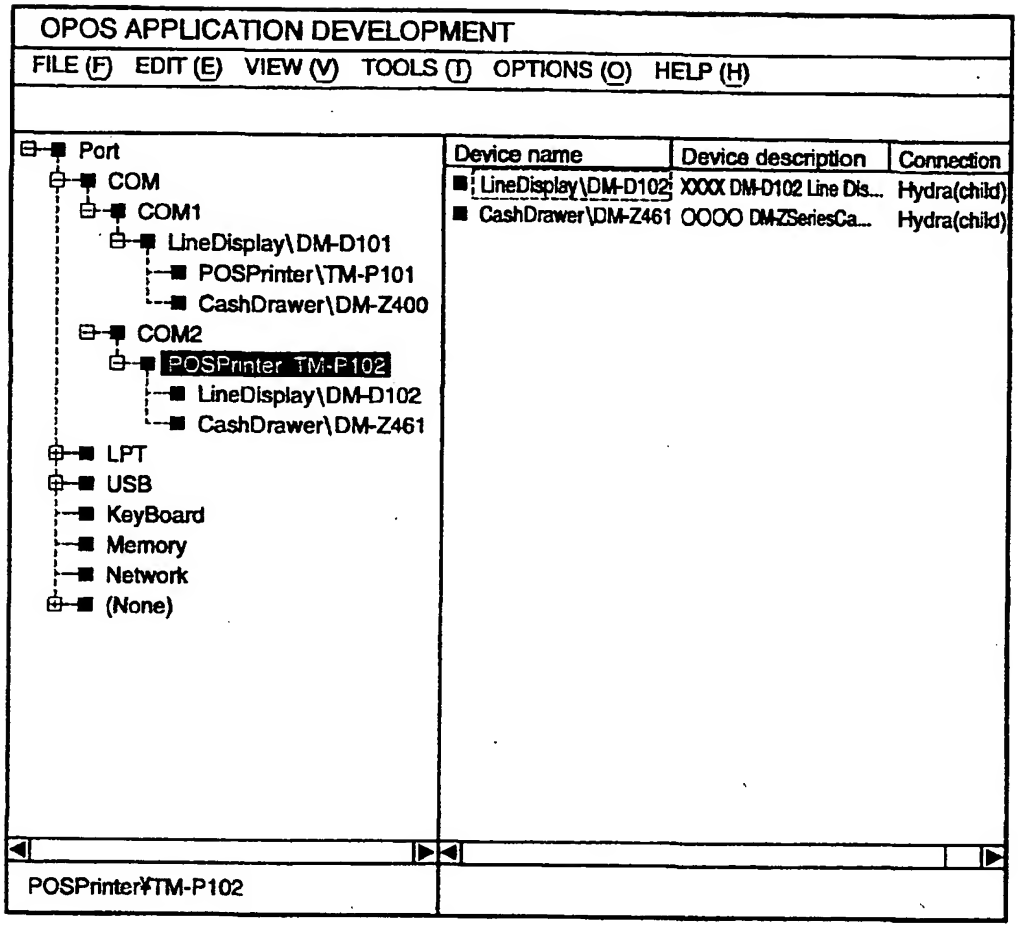

OPOS APPLICATION DEVELOPMENT
FILE (F) EDIT (E) VIEW (V) TOOLS (T) OPTIONS (O) HELP (H)
Device
POSPrinter
TM-P101
LineDisplay
DM-D101
CashDrawer
DM-Z400
MICR
MSR
KeyLock
Scanner
HardTotals
CashChanger
CoinDispenser
POSKeyboard
fjkbdso
Scale
SignatureCapture
ToneIndicator
BumpBar
FiscalPrinter
PINPad
RemoteOrderDisplay
DM-Z400
CashDrawer

FIG. 11

(a)


OPOS APPLICATION DEVELOPMENT
FILE (F) EDIT (E) VIEW (V) TOOLS (T) OPTIONS (O) HELP (H)
Port
COM
COM1
LineDisplay\DM-D101
POSPrinter\TM-P101
CashDrawer\DM-Z400
COM2
POSPrinter TM-P102
LineDisplay\DM-D102
CashDrawer\DM-Z461
LPT
USB
KeyBoard
Memory
Network
(None)
Device name
Device description
Connection
LineDisplay\DM-D102
XXXX DM-D102 Line Dis...
Hydra(child)
CashDrawer\DM-Z461
OOOO DM-ZSeriesCa...
Hydra(child)
POSPrinter¥TM-P102

(b)

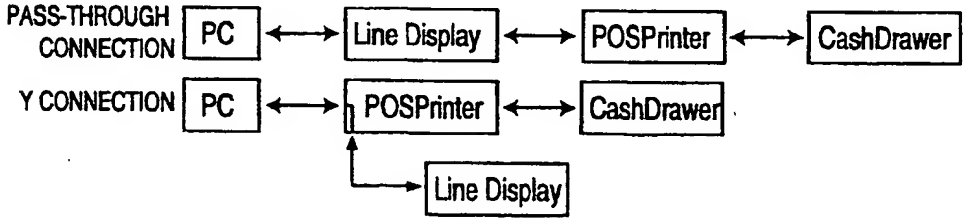
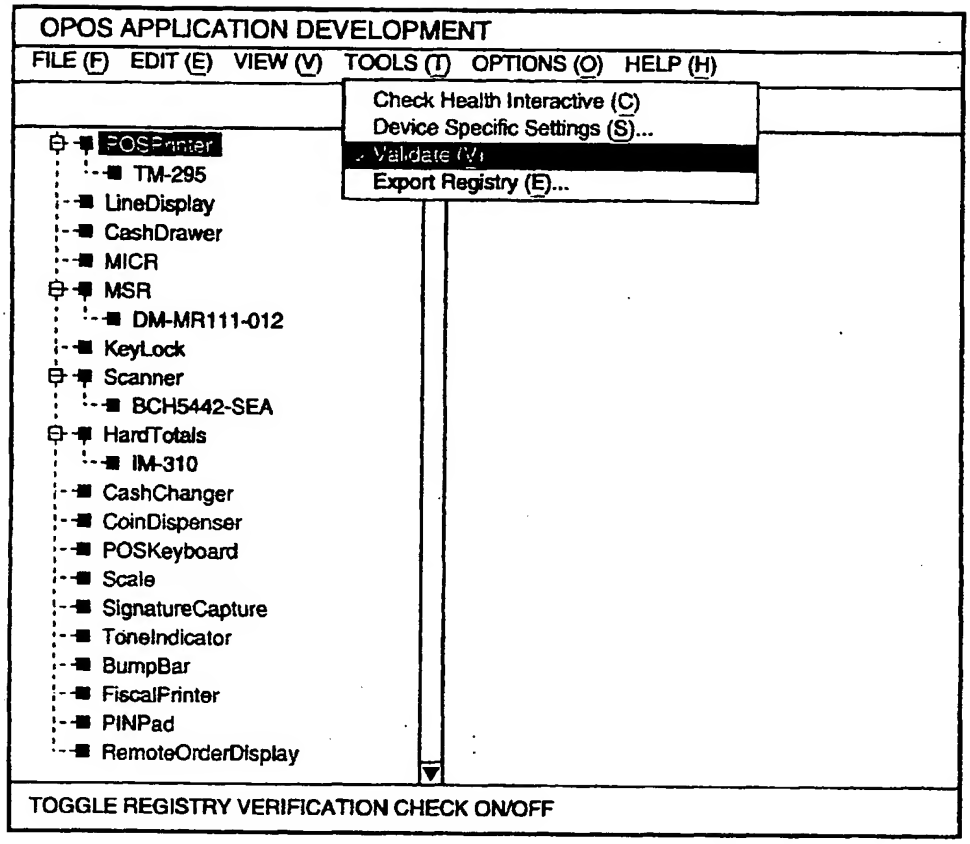

PASS-THROUGH CONNECTION
PC
Line Display
POSPrinter
CashDrawer
Y CONNECTION
PC
POSPrinter
CashDrawer
Line Display

FIG. 12

(a)


OPOS APPLICATION DEVELOPMENT
FILE (F) EDIT (E) VIEW (V) TOOLS (T) OPTIONS (O) HELP (H)
Check Health Interactive (C)
Device Specific Settings (S)...
Validate (V)
Export Registry (E)...
POSPrinter
TM-295
LineDisplay
CashDrawer
MICR
MSR
DM-MR111-012
KeyLock
Scanner
BCH5442-SEA
HardTotals
IM-310
CashChanger
CoinDispenser
POSKeyboard
Scale
SignatureCapture
ToneIndicator
BumpBar
FiscalPrinter
PINPad
RemoteOrderDisplay
TOGGLE REGISTRY VERIFICATION CHECK ON/OFF

(b)

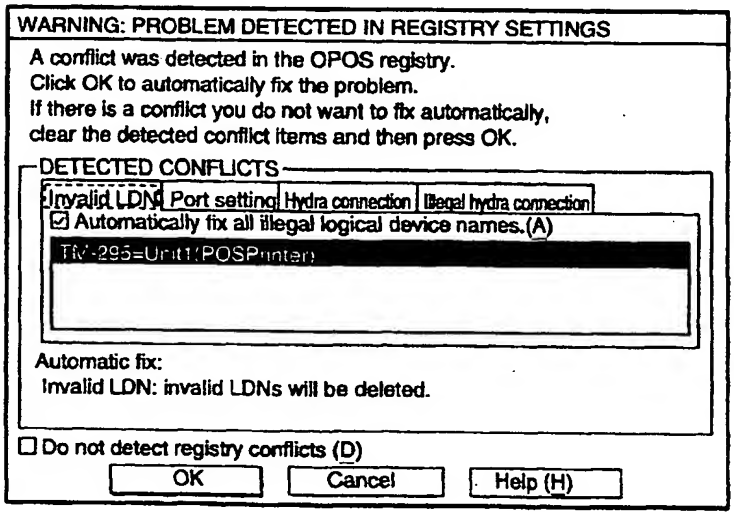
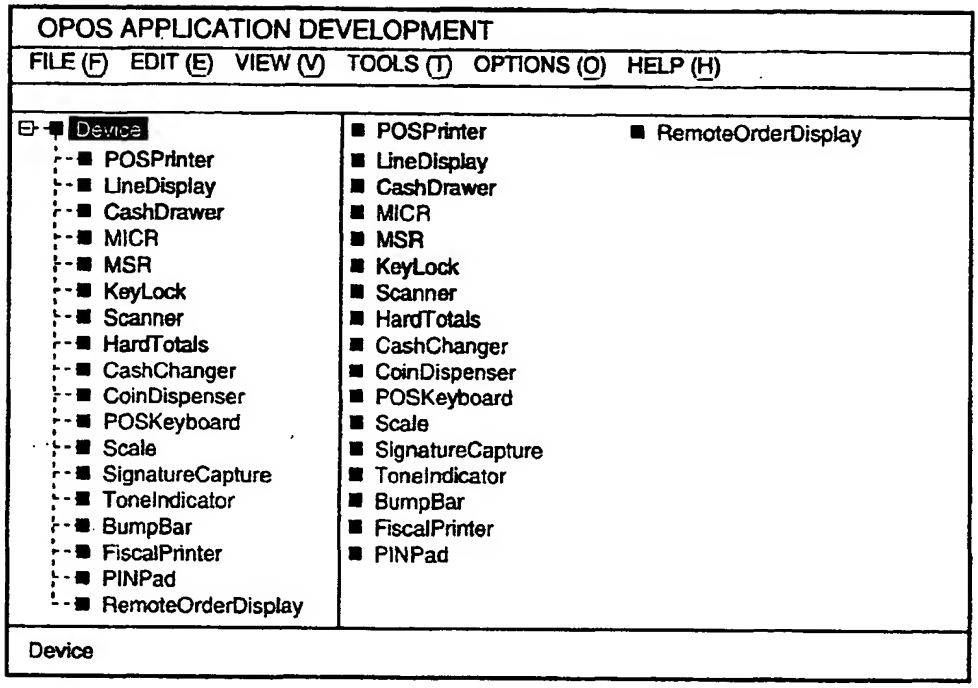

WARNING: PROBLEM DETECTED IN REGISTRY SETTINGS
A conflict was detected in the OPOS registry.
Click OK to automatically fix the problem.
If there is a conflict you do not want to fix automatically,
clear the detected conflict items and then press OK.
DETECTED CONFLICTS
Invalid LDN
Port setting
Hydra connection
Illegal hydra connection
Automatically fix all illegal logical device names.(A)
TM-295=Unit1:POSPrinter
Automatic fix:
Invalid LDN: invalid LDNs will be deleted.
Do not detect registry conflicts (D)
OK
Cancel
Help (H)

FIG. 13

(a)


OPOS APPLICATION DEVELOPMENT
FILE (F) EDIT (E) VIEW (V) TOOLS (T) OPTIONS (O) HELP (H)
Device
POSPrinter
LineDisplay
CashDrawer
MICR
MSR
KeyLock
Scanner
HardTotals
CashChanger
CoinDispenser
POSKeyboard
Scale
SignatureCapture
ToneIndicator
BumpBar
FiscalPrinter
PINPad
RemoteOrderDisplay
POSPrinter
LineDisplay
CashDrawer
MICR
MSR
KeyLock
Scanner
HardTotals
CashChanger
CoinDispenser
POSKeyboard
Scale
SignatureCapture
ToneIndicator
BumpBar
FiscalPrinter
PINPad
RemoteOrderDisplay
Device

(b)

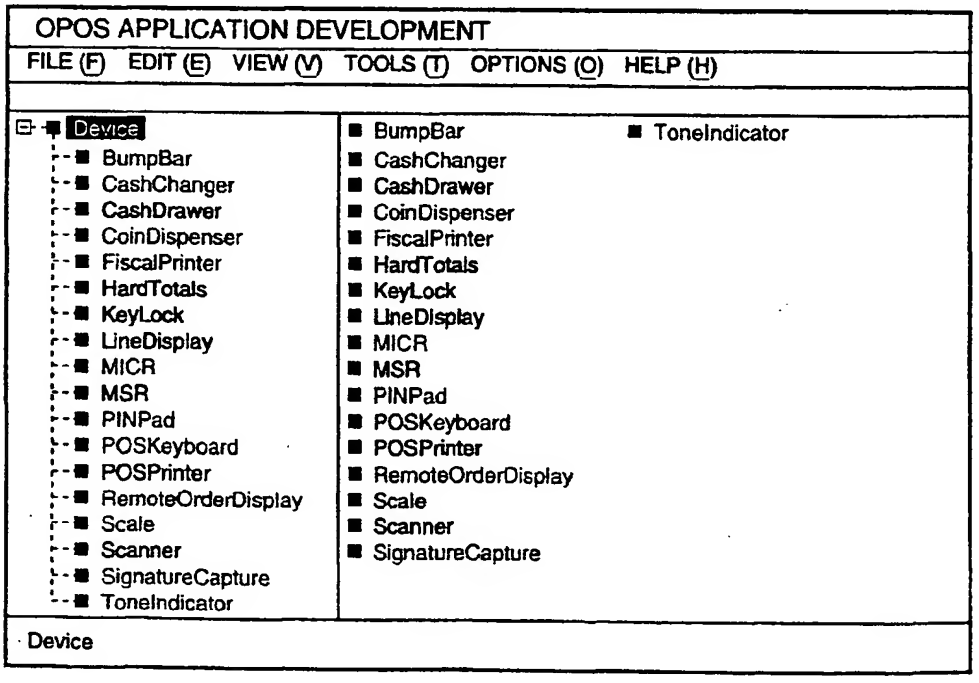

OPOS APPLICATION DEVELOPMENT
FILE (F) EDIT (E) VIEW (V) TOOLS (T) OPTIONS (O) HELP (H)
Device
BumpBar
CashChanger
CashDrawer
CoinDispenser
FiscalPrinter
HardTotals
KeyLock
LineDisplay
MICR
MSR
PINPad
POSKeyboard
POSPrinter
RemoteOrderDisplay
Scale
Scanner
SignatureCapture
ToneIndicator
BumpBar
CashChanger
CashDrawer
CoinDispenser
FiscalPrinter
HardTotals
KeyLock
LineDisplay
MICR
MSR
PINPad
POSKeyboard
POSPrinter
RemoteOrderDisplay
Scale
Scanner
SignatureCapture
ToneIndicator
Device

FIG. 14

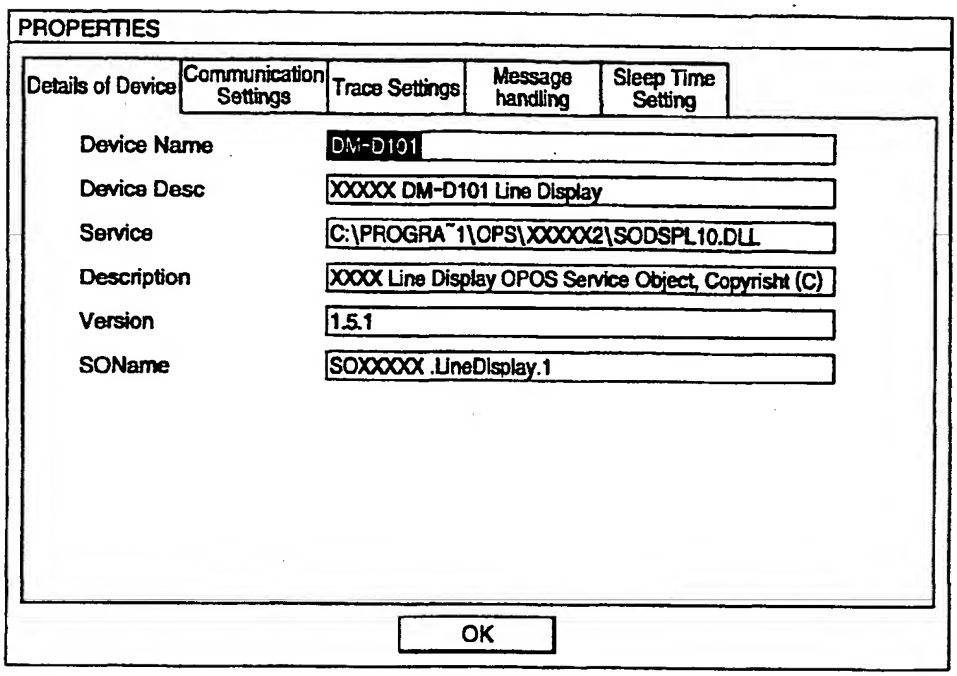

PROPERTIES
Details of Device
Communication Settings
Trace Settings
Message handling
Sleep Time Setting
Device Name
DM-D101
Device Desc
XXXXX DM-D101 Line Display
Service
C:\PROGRA~1\OPS\XXXXX2\SODSPL10.DLL
Description
XXXX Line Display OPOS Service Object, Copyrisht (C)
Version
1.5.1
SOName
SOXXXXX .LineDisplay.1
OK

FIG. 15

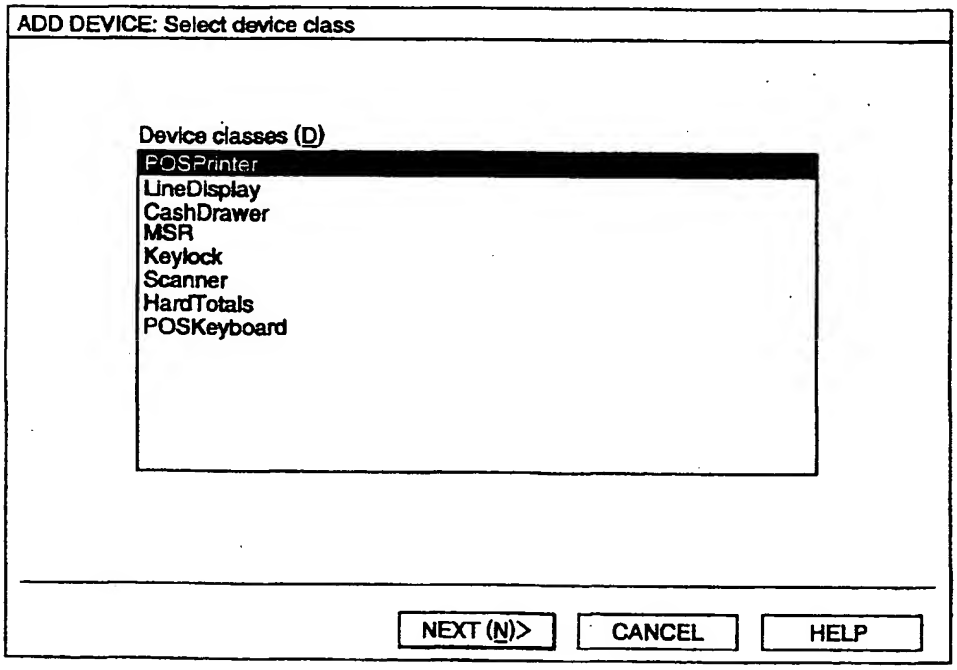

ADD DEVICE: Select device class
Device classes (D)
POSPrinter
LineDisplay
CashDrawer
MSR
Keylock
Scanner
HardTotals
POSKeyboard
NEXT (N)>
CANCEL
HELP

FIG. 16

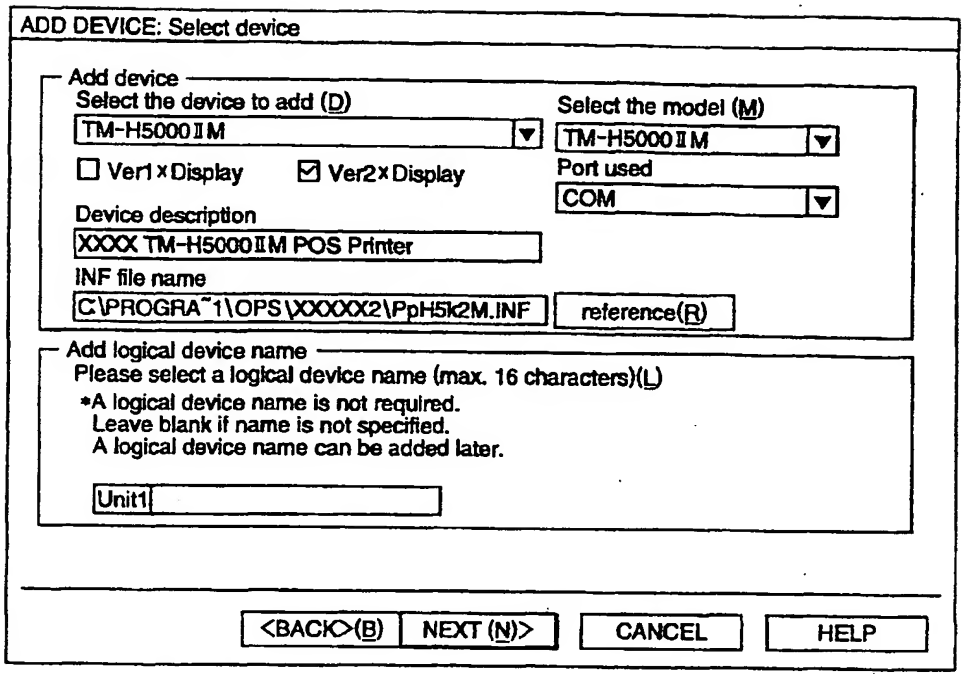

ADD DEVICE: Select device
Add device
Select the device to add (D)
TM-H5000IIM
Select the model (M)
TM-H5000IIM
Ver1xDisplay
Ver2xDisplay
Port used
COM
Device description
XXXX TM-H5000IIM POS Printer
INF file name
C:\PROGRA~1\OPS\XXXXX2\PpH5k2M.INF
reference(R)
Add logical device name
Please select a logical device name (max. 16 characters)(L)
*A logical device name is not required.
Leave blank if name is not specified.
A logical device name can be added later.
Unit1
<BACK>(B)
NEXT (N)>
CANCEL
HELP

FIG. 17

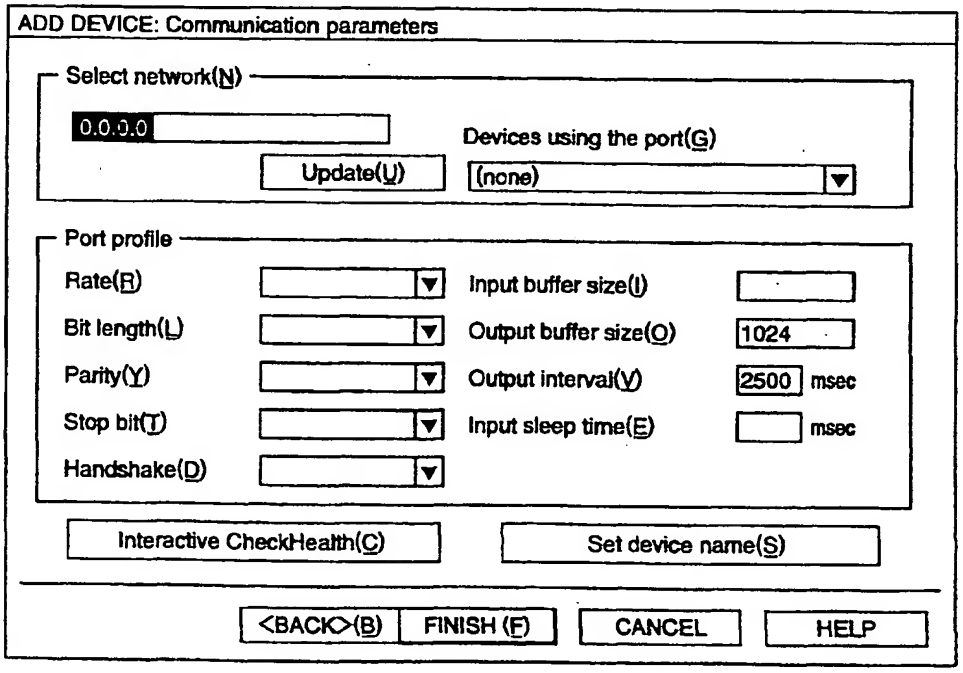

ADD DEVICE: Communication parameters
Select network(N)
0.0.0.0
Update(U)
Devices using the port(G)
(none)
Port profile
Rate(R)
Bit length(L)
Parity(Y)
Stop bit(T)
Handshake(D)
Input buffer size(I)
Output buffer size(O)
1024
Output interval(V)
2500
msec
Input sleep time(E)
msec
Interactive CheckHealth(C)
Set device name(S)
<BACK>(B)
FINISH (F)
CANCEL
HELP

FIG. 18

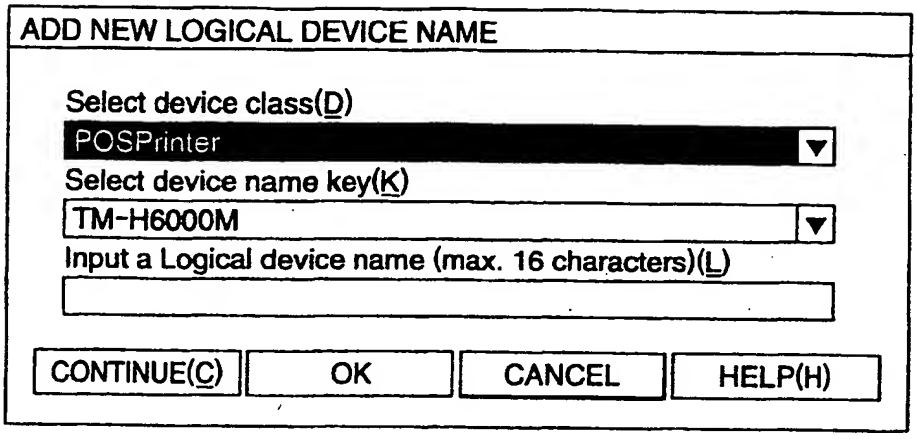

ADD NEW LOGICAL DEVICE NAME
Select device class(D)
POSPrinter
Select device name key(K)
TM-H6000M
Input a Logical device name (max. 16 characters)(L)
CONTINUE(C)
OK
CANCEL
HELP(H)

FIG. 19

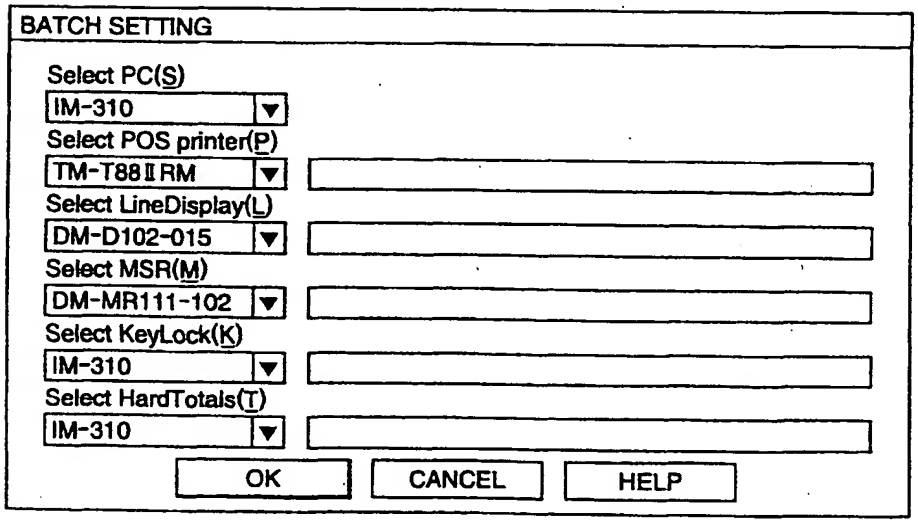

BATCH SETTING
Select PC(S)
IM-310
Select POS printer(P)
TM-T88II RM
Select LineDisplay(L)
DM-D102-015
Select MSR(M)
DM-MR111-102
Select KeyLock(K)
IM-310
Select HardTotals(T)
IM-310
OK
CANCEL
HELP

FIG. 20

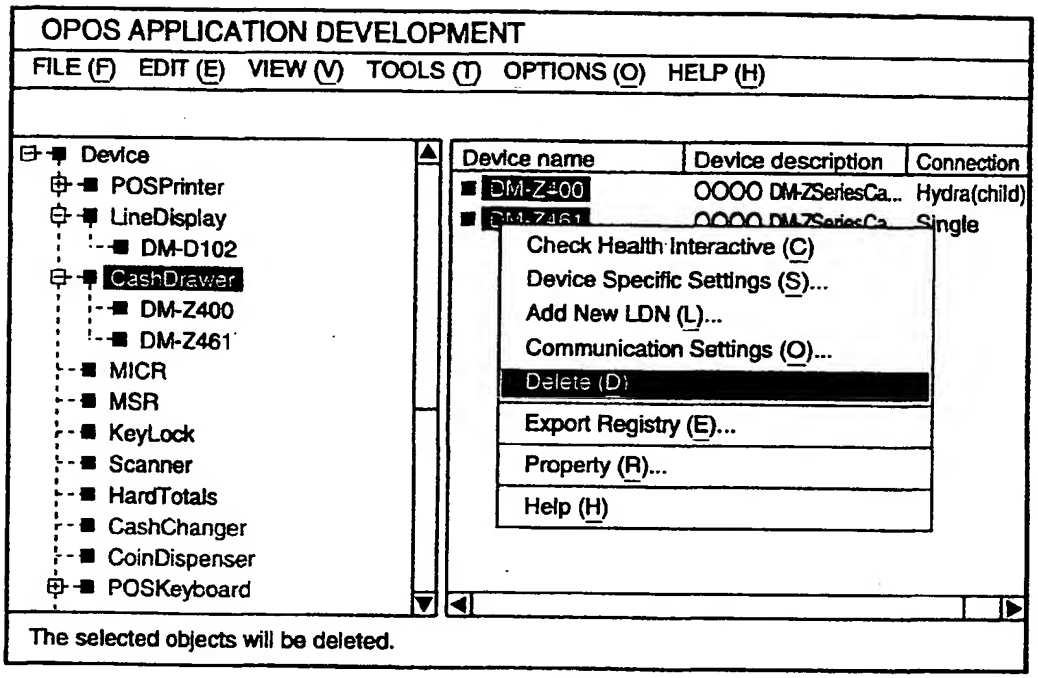

OPOS APPLICATION DEVELOPMENT
FILE (F) EDIT (E) VIEW (V) TOOLS (T) OPTIONS (O) HELP (H)
Device
POSPrinter
LineDisplay
DM-D102
CashDrawer
DM-Z400
DM-Z461
MICR
MSR
KeyLock
Scanner
HardTotals
CashChanger
CoinDispenser
POSKeyboard
Device name
Device description
Connection
DM-Z400
OOOO DM-ZSeriesCa...
Hydra(child)
Single
Check Health Interactive (C)
Device Specific Settings (S)...
Add New LDN (L)...
Communication Settings (O)...
Delete (D)
Export Registry (E)...
Property (R)...
Help (H)
The selected objects will be deleted.

FIG. 21

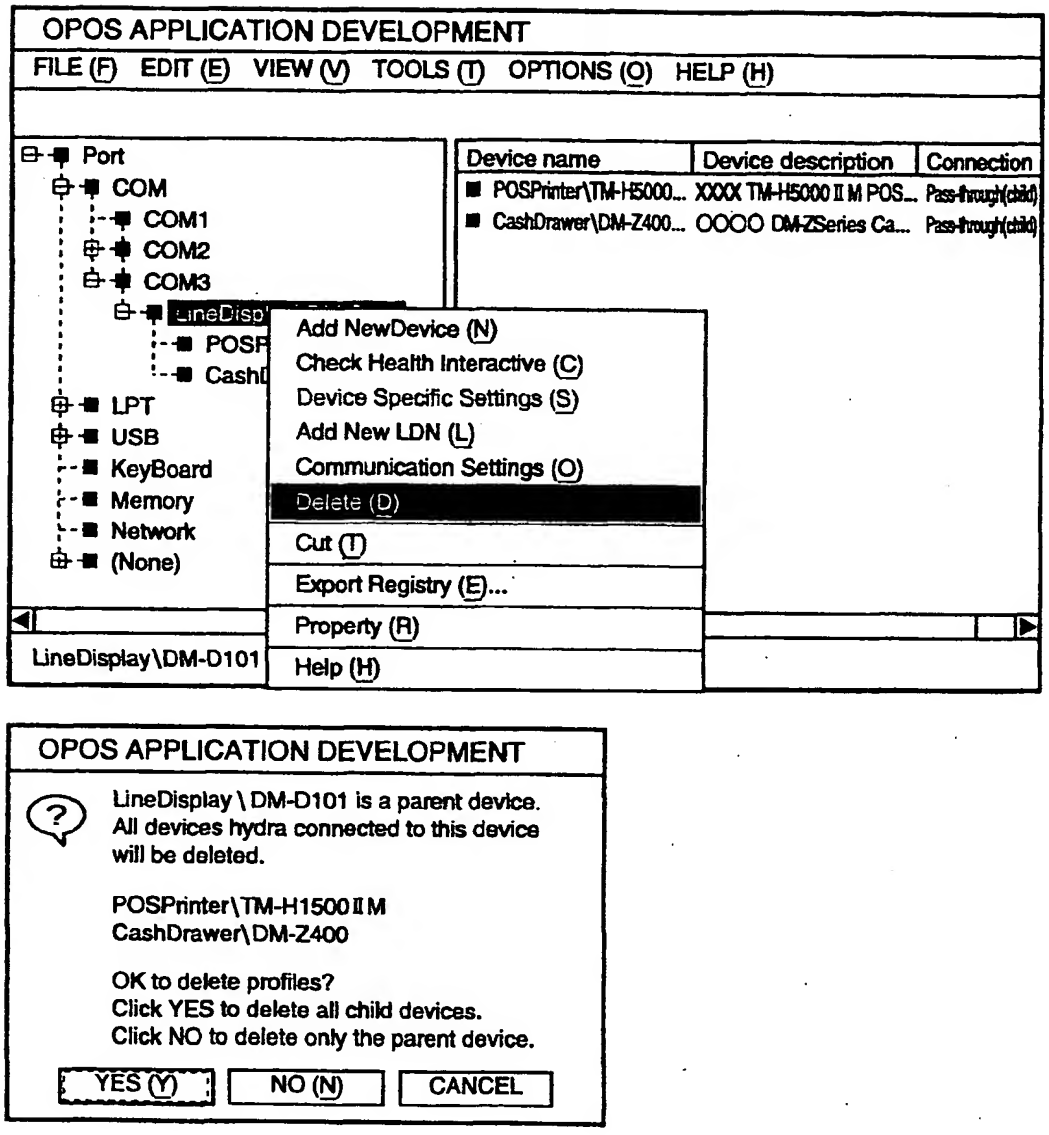
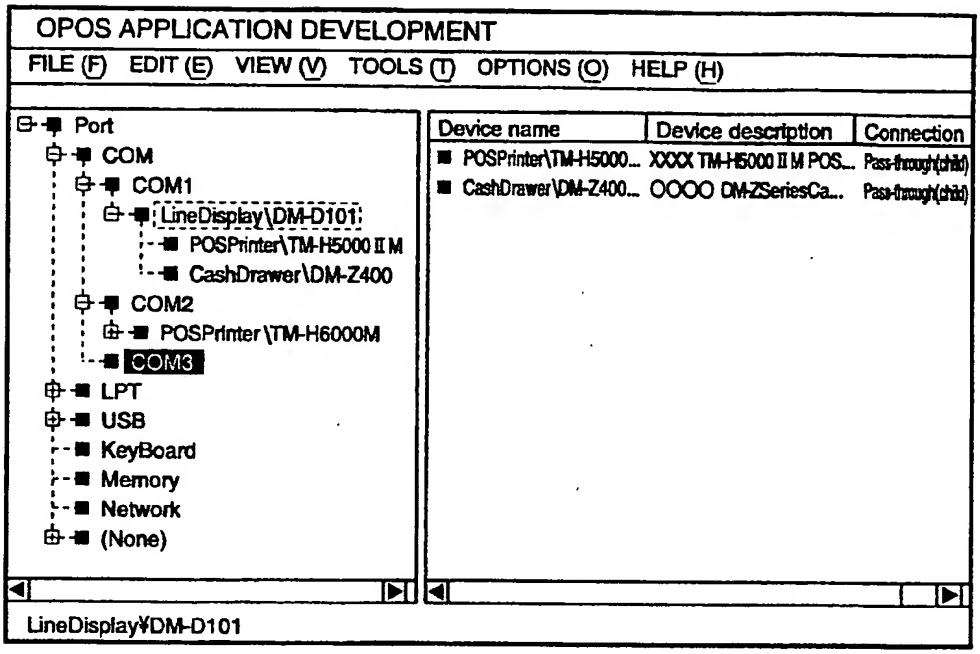

OPOS APPLICATION DEVELOPMENT
FILE (F) EDIT (E) VIEW (V) TOOLS (T) OPTIONS (O) HELP (H)
Port
COM
COM1
COM2
COM3
LPT
USB
KeyBoard
Memory
Network
(None)
Device name
Device description
Connection
POSPrinter\TM-H5000...
XXXX TM-H5000 II M POS...
CashDrawer\DM-Z400...
OOOO DM-ZSeries Ca...
Add NewDevice (N)
Check Health Interactive (C)
Device Specific Settings (S)
Add New LDN (L)
Communication Settings (O)
Delete (D)
Cut (T)
Export Registry (E)...
Property (R)
Help (H)
LineDisplay\DM-D101
OPOS APPLICATION DEVELOPMENT
LineDisplay \ DM-D101 is a parent device.
All devices hydra connected to this device
will be deleted.
POSPrinter\TM-H1500 II M
CashDrawer\DM-Z400
OK to delete profiles?
Click YES to delete all child devices.
Click NO to delete only the parent device.
YES (Y)
NO (N)
CANCEL

FIG. 22

(a)


OPOS APPLICATION DEVELOPMENT
FILE (F) EDIT (E) VIEW (V) TOOLS (T) OPTIONS (O) HELP (H)
Port
COM
COM1
LineDisplay\DM-D101
POSPrinter\TM-H5000 II M
CashDrawer\DM-Z400
COM2
POSPrinter\TM-H6000M
COM3
LPT
USB
KeyBoard
Memory
Network
(None)
Device name
Device description
Connection
POSPrinter\TM-H5000...
XXXX TM-H5000 II M POS...
Pass-through(child)
CashDrawer\DM-Z400...
OOOO DM-ZSeriesCa...
Pass-through(child)
LineDisplay¥DM-D101

(b)

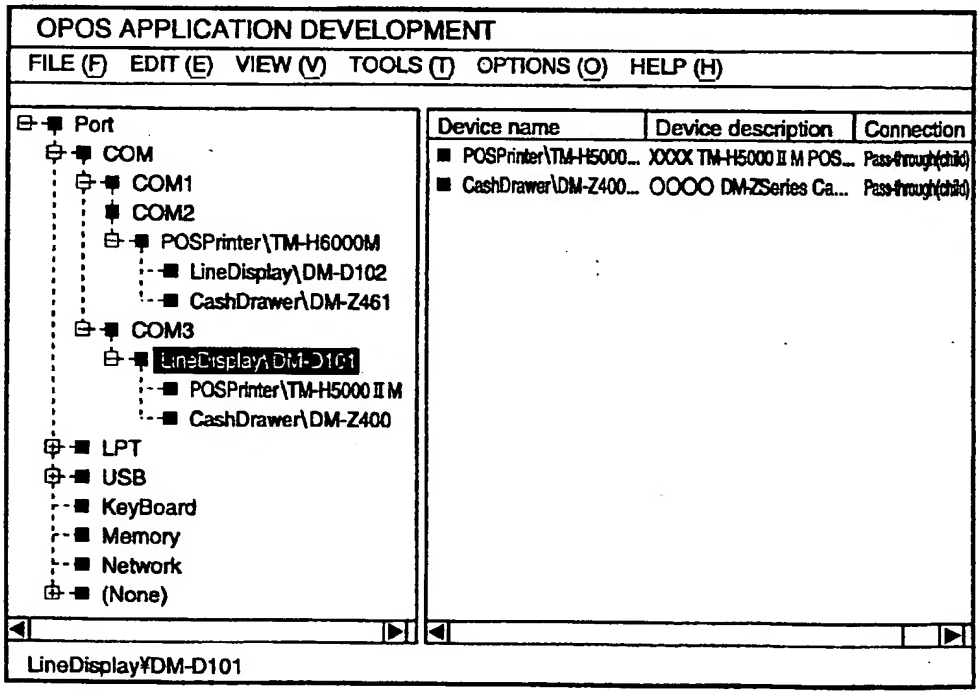

OPOS APPLICATION DEVELOPMENT
FILE (F) EDIT (E) VIEW (V) TOOLS (T) OPTIONS (O) HELP (H)
Port
COM
COM1
COM2
POSPrinter\TM-H6000M
LineDisplay\DM-D102
CashDrawer\DM-Z461
COM3
LineDisplay\DM-D101
POSPrinter\TM-H5000 II M
CashDrawer\DM-Z400
LPT
USB
KeyBoard
Memory
Network
(None)
Device name
Device description
Connection
POSPrinter\TM-H5000...
XXXX TM-H5000 II M POS...
Pass-through(child)
CashDrawer\DM-Z400...
OOOO DM-ZSeries Ca...
Pass-through(child)
LineDisplay¥DM-D101

FIG. 23